

# Combinatoria como conteo

## Contenido

- 2.1. usando un arbol
- 2.2. Permutaciones y combinaciones
- 2.3. El principio de exclusión-inclusión
- 2.4. La fórmula de Stirling

Nuestro Cerebro Discreto debe ser capaz de **contar objetos**, es decir, debe poder responder preguntas como:

- ¿Cuántos subconjuntos hay en un conjunto de tamaño  $n$ ?
- ¿Cuántos subconjuntos de los  $k$  elementos están en un conjunto de tamaño  $n$ ?
- ¿Cuántas funciones inyectivas pueden asignar dos conjuntos  $A$  y  $B$  con tallas  $m$  y  $n$ ?

Las preguntas anteriores se denominan **tareas de conteo**, en la medida en que se pueden descomponer en tareas más simples. Consideremos, por ejemplo, la última tarea. tenemos dos conjuntos  $A$  y  $B$  con tallas  $m$  y  $n$ , donde  $m \leq n$ . Entonces:

- Estamos considerando funciones inyectivas (uno a uno).  $f : X \rightarrow Y$ , es decir, la referencia es  $X$  (razonamos de izquierda a derecha).
- en cada función  $f$ , cada elemento  $x_i \in X$  debe asignarse *a uno y sólo un elemento*  $y_j \in Y$  *y a ninguno de ellos*.

Por ejemplo, en **setfunction** tenemos  $X = \{a, b, c\}$  y  $Y = \{1, 2, 3, 4\}$ , es decir  $m = 3$  y  $n = 4$ . El mapeo, digamos  $f$ , es inyectivo ya que: (a) cada elemento en  $X$  tiene una **imagen** y solo una imagen en  $Y$ , (b) ningún elemento en  $Y$  es la imagen de más de un elemento en  $X$ .

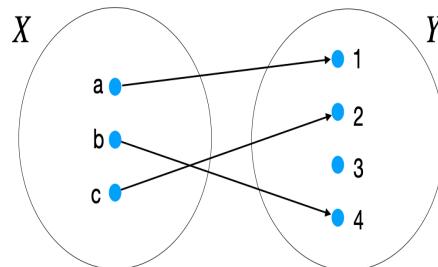


Fig. 2.1 Función uno a uno entre dos conjuntos.

## 2.1. Usando un árbol

Cualquier tarea de conteo puede representarse mediante un **árbol**. Los árboles son estructuras clave en nuestro cerebro discreto, porque codifican una jerarquía. En particular, el **árbol enraizado** (ver **rtree**) puede verse como *la expansión recursiva de arriba hacia abajo de una raíz a través de varios niveles hasta alcanzar las hojas*:

[Saltar al contenido principal](#)

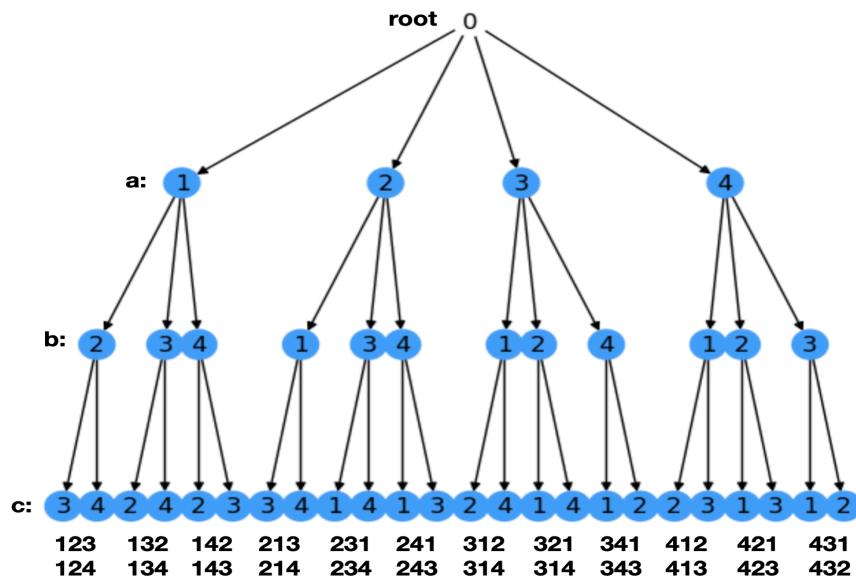


Fig. 2.2 Árbol con raíces donde las hojas son la solución a un problema de conteo.

En particular, tenemos:

- La **raíz es el nodo** padre superior del árbol y abarca varios **hijos**, formando así el primer nivel de **hermanos**.
- Cada hijo es una **hoja** (un nodo sin hijos) o el parente de varios **hijos**, formando así el segundo nivel de **hermanos**.
- El proceso continúa hasta que todos los nuevos hermanos se van.

En nuestro problema de conteo, utilizamos un árbol con raíces para descomponer la tarea de conteo original: *¿cuántas funciones uno a uno tenemos entre dos conjuntos de  $m$  elementos,  $m \leq n$ ?*

- Cada **nivel** en el árbol (raíz no incluida) codifica una posible elección para cada uno de los  $m$  variables. En la figura,  $m = 3$  y denotamos  $x_1$  por  $a$ ,  $x_2$  por  $b$  y  $x_3$  por  $c$ . Por lo tanto, en el primer nivel tenemos  $n = 4$  opciones para  $a: \{1, 2, 3, 4\}$  (los hijos de la raíz).
- En este problema no importa ni el orden de selección de las variables  $a, b, c$  ni el de las asignaciones. Simplemente seguimos un **orden sistemático**: un orden creciente tanto para las variables ( $a, b, c$ ) y tareas ( $1, 2, 3, 4$ ). Sin embargo, el árbol se puede construir/visitar/atravesar de tres maneras diferentes. Para comprender mejor estas formas, considere que un árbol es una estructura **recursiva**, es decir, está compuesta de **subárboles**. Por ejemplo, la raíz se compone de cuatro subárboles  $T_1, T_2, T_3, T_4$  cuyas raíces son las cuatro posibles asignaciones para la variable  $a$ . Entonces los **recorridos** son:
  - *Hacer un pedido*. Primero visita la raíz. Luego visite el primer subárbol (izquierdo)  $T_1$  de la raíz en preorden antes de visitar el segundo subárbol  $T_2$ .
  - *En orden*. Primero visite el primer subárbol (izquierdo)  $T_1$  de la raíz en orden antes de visitar la raíz misma. Después de visitar la raíz, proceda a visitar  $T_2$  en orden, etc.
  - *Orden de publicación*. Primero visite el primer subárbol (izquierdo)  $T_1$  de la raíz en postorden. Luego, proceda a visitar  $T_2$  en postorden, etc. Finalmente, visita la raíz.
- Sin embargo, existen dos recorridos principales para resolver problemas de IA:
  - *Profundidad primero primero* (DFS). Este es básicamente *cualquiera de los recorridos anteriores*. Después del pedido anticipado, por ejemplo, construimos **rutas** (secuencias de nodos) como **123**. En realidad, este es el primer camino explorado en preorden. Entonces sigue **124, 132, 134 ... hasta 431 y 432**. Vea las rutas (sin incluir la raíz) en la parte inferior de **perm**.
  - *La respiración primero, primero* (BFS). Dada la raíz del árbol, visite la raíz de cada uno de sus subárboles (p. ej.  $T_1, T_2, T_3, T_4$ ). Luego visite las raíces de todos los subárboles de  $T_1$  etc. Lo único que hacemos es visitar el nivel  $l$  antes de visitar el nivel  $l + 1$ . En BFS, *todos los caminos se expanden gradualmente* hasta llegar a las hojas. Por ejemplo a nivel 1 (sin contar la raíz) tenemos 4 caminos incompletos: **1 \* \*, 2 \* \*, 3 \* \*, 4 \***. Luego expandimos los subárboles de izquierda a derecha y para el primer subree tenemos: **12\*, 13\*, 14\***. Finalmente, en las hojas tenemos **123, 124, 132, 134 ... hasta 431 y 432**.

[Saltar al contenido principal](#)

## 2.2. Permutaciones y combinaciones

### 2.2.1. Vinculación de permutaciones y combinaciones

Las rutas antes mencionadas no tienen valores de nodos repetidos y, por lo tanto, son soluciones al problema de conteo. Por ejemplo el camino  $\pi = 1\mathbf{2}4$  significa una función  $f_\pi$  donde  $a = 1, b = 2, c = 4$  y nos fuimos  $d$ no asignado. Por lo tanto, la tarea de contar se descompone en  $m = 3$  contar tareas (una por nivel): en el primer nivel podemos elegir  $n$  valores para  $a$ ; dados esos valores, solo podemos elegir  $n - 1$  valores para  $b$  consecuentemente  $n - 2$  valores para  $c$ . Como resultado, el número total de funciones uno a uno entre  $X$  y  $Y$  viene dada por la **regla del producto**  $n(n - 1)(n - 2)$  y este número está dado por el  $P(n, r)$

**r-permutación :**

$$P(n, r) = n(n - 1) \cdots (n - r + 1),$$

donde en este caso  $r = m$  (en general  $m \leq n$ ). Curiosamente, para  $m = n$  (niveles en el árbol) tenemos la **permutación clásica**  $P(n, n)$  o simplemente  $P(n)$ :

$$P(n) = n! = n(n - 1)(n - 2) \cdots 1!$$

dónde! es el factorial y  $0! = 1$ . En combinatoria, las permutaciones codifican todos los órdenes posibles de  $n$  elementos distintos . En realidad, es sencillo demostrar que

$$P(n, r) = \frac{n!}{(n - r)!}.$$

En otras palabras, las permutaciones r eliminan  $(n - r)$  productos (niveles en el árbol) del total  $n!$  permutaciones . Por ejemplo, en el ejemplo de **setfunction** solo necesitamos agregar 1 elemento ( $n - m = 1$ ) a  $X$  para hacer  $m = n$ . Esto resultará en 1 más nivel en el árbol en **rtree**.

**Combinaciones** . ¿Qué pasa si en el ejemplo anterior el orden no importa? Por ejemplo, caminos **124, 142, 214, 241, 412 y 421** se consideran *iguales* porque básicamente contienen los mismos nodos 1, 2, 4 en *cualquier orden posible* . Esto significa que hay muchos grupos de caminos en el árbol de **rtree** .

Una **combinación** r cuenta cuántos grupos de  $r$  elementos se pueden obtener de  $n \geq r$ . Se define como el número  $C(n, r)$ :

$$C(n, r) = \binom{n}{r} = \frac{n!}{r!(n - r)!}.$$

Por lo tanto, si analizamos la definición de  $P(n, r)$  queda claro que  $C(n, r)$  elimina  $r!(n - r)!$  del numerador  $n!$ . Este es un buen ejemplo de la llamada **regla de división** : hay  $k/d$  formas de realizar una tarea si se puede realizar en  $k$  formas y para cada una de estas formas exactamente  $d$  de las formas se consideran iguales. En  $C(n, r)$ , tenemos eso  $k = n!/r!(n - r)! = P(n, r)$  y  $d = r! = P(r)$  ya que cualquiera de los  $P(n, r)$  Los caminos en el árbol se pueden hacer en  $r!$  maneras si el orden de los nodos no importa. Como resultado:

$$C(n, r) = \frac{P(n, r)}{P(r)} = \frac{\frac{n!}{(n - r)!}}{r!} = \frac{n!}{r!(n - r)!}.$$

### 2.2.2. Permitir repeticiones

En nuestro ejemplo motivador (encontrar inyecciones  $f : X \rightarrow Y$ ), cada elemento de la  $Y$  El conjunto se puede seleccionar *una vez* en cada función uno a uno diferente. Consideremos permitir la repetición de alguno o algunos de los elementos. Esto conduce a problemas de conteo más realistas (por ejemplo, repetir letras para formar códigos, objetos de la misma variedad, etc.).

Supongamos, por ejemplo, que en lugar de enumerar las funciones inyectivas (uno a uno) entre  $X$  y  $Y$ , preguntamos *cuantas funciones*  $f : X \rightarrow Y$  (uno a uno o no) existen. La respuesta es simple una vez que nos damos cuenta de que cualquier elemento en  $X$  se puede asignar (sólo una vez, dada la definición de una función ) a  $y$  de los elementos de  $Y$ . Esta consideración nos permite descomponer la tarea de contar de la siguiente manera:

[Saltar al contenido principal](#)

- Consideremos primero el elemento  $a \in X$  en **setfunction**. Se puede asignar a cualquier elemento  $1, 2, 3, 4$  de  $Y$ . Es decir, por  $n$  tenemos  $n$  posibles asignaciones.
- De manera similar, para  $b \in Y$  tenemos el mismo número de tareas:  $n$ .
- como tenemos  $m$  elementos en  $X$ , el número total de asignaciones (funciones) es  $n \cdot n \cdots n$  ( $m$  veces), es decir  $n^m$ .
- En términos de una representación basada en árbol, tendremos un  $n$ -árbol ario ( $n$  sucursales por nivel) con  $m$  niveles (sin contar la raíz). Este árbol tiene  $n^m$  hojas.

Por lo tanto, una **r-permutación con repetición** es  $P_\sigma(n, r) = n^r$ .

A veces, encontrar una  $P_\sigma(n, r)$  no es tan sencillo y necesita un esquema de mapeo o **codificación previo** (esto será útil más adelante para demostrar que un problema es NP-hard).

Por ejemplo, [para encontrar todos los subconjuntos posibles de un conjunto finito](#)  $S = \{1, 2, \dots, n\}$  (el tamaño del **conjunto de potencia**  $\mathcal{P}^S$ ), comenzamos a enumerar los elementos:

- Obviamente  $\emptyset$  pertenece a  $\mathcal{P}^S$ . Estos conjuntos son respectivamente el subconjunto con  $0$  elementos y que con todos los  $n$  elementos.
- A continuación, consideramos los subconjuntos con  $1$  elemento: son  $\{1\}, \{2\}, \dots, \{n\}$ , es decir  $n$ .
- Luego, consideramos los subconjuntos con  $2$  elementos. Como obviamente el orden no importa, es decir  $\{i, j\} = \{j, i\}$ , tenemos una **combinación r**:  $\binom{n}{2}$ .
- De manera similar para contar los subconjuntos con  $3, 4, \dots, n - 1$  elementos que tenemos  $\binom{n}{3}, \binom{n}{4}, \dots, \binom{n}{n-1}$ .

Desde  $1 = \binom{n}{0} = \binom{n}{n}$  y  $n = \binom{n}{1}$ , el tamaño del conjunto de potencias es ([usamos la notación](#)  $| \cdot |$  para indicar el tamaño de algo):

$$|\mathcal{P}^S| = \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n} = 2^n .$$

Aquí aplicamos la **regla de la suma**: si hay  $n$  tareas de conteo independientes, considerando el número de formas cada uno, el número total de formas es  $\sum_{i=1}^n s_i$ .

Sin embargo, no es tan obvio que la suma de r-combinaciones sea  $2^n$ . Este punto lo aclararemos más adelante, al estudiar el **teorema del Binomio**.

Una alternativa mucho más simple (e inteligente) proviene de [codificar cada subconjunto como una cadena binaria de n bits](#). Por ejemplo,  $\emptyset = 000 \dots 0$ , mientras  $S = 111 \dots 1$ . En el medio tenemos que todos los conjuntos con  $2$  Los elementos están codificados por cadenas binarias de longitud  $n$  con solo  $2$  bits "activados" (establecidos en  $1$ s). Lo mismo ocurre con los conjuntos de longitudes  $3 \dots n - 1$ . Como resultado, existe una *biyección* entre  $\mathcal{P}^S$  y los números binarios de  $n$  bits.

Obviamente, el número de cadenas diferentes con  $n$  bits es  $2^n$ . En otras palabras, tenemos dos elementos  $\{0, 1\}$  que puede repetirse en la medida en que el número total de elementos sean  $n$ . Más formalmente, tenemos que encontrar todas las funciones  $S \rightarrow B$  entre  $S = \{1, 2, \dots, n\}$  y el conjunto base  $B = \{0, 1\}$ , es decir  $P_\sigma(2, n) = 2^n$ .

El resultado anterior se puede generalizar a cualquier otra base como  $a^n$ . Por ejemplo, el número de códigos [Navilens](#) es  $4^n$ , donde  $n$  es el número de celdas, ya que estos códigos tienen cuatro colores: cian, magenta, amarillo y negro (el borde blanco sólo facilita la localización automática del marco circundante). De hecho,  $4^n = (2 \cdot 2)^n \gg 2^n$  y es por eso que los códigos Navilens son mucho más densos que los códigos QR (incluso descontando los bits de la suma de control), es decir, podemos representar muchos más códigos con menos bits.

## 2.2.3. Elementos indistinguibles

Hasta ahora, los elementos del problema de contar se pueden repetir. Sin embargo, algunos problemas de conteo se basan en considerar **tipos de elementos**, en lugar de elementos en sí. En otras palabras, [en algunos problemas de conteo no importa el número de elementos que satisfacen tal o cual propiedad](#), es más importante considerar los elementos de tipo  $a, q$  elementos de tipo  $b, p$ , etc, y luego contabilizando el número total de formas. Esto es como imponer  $p$  ceros y  $q$  unos en una cuerda con  $n$  bytes, donde  $p + q = n$ .

**Problemas de fuente**. Consideremos el caso en el que tenemos  $n$  fuentes (tipos), y tenemos  $r \leq n$  formas de seleccionarlos, donde se permiten repeticiones. Por ejemplo, dada  $n$  (elementalmente ilimitados) colores, solo tenemos  $r$  formas de seleccionarlos (tal vez porque nuestra paleta de colores es ciega a más de colores).

[Saltar al contenido principal](#)

### 2.2.3.1. Permutaciones con repetición

Si el orden importa, tenemos que aclarar ahora que se permiten muchas formas para cada tipo. Por ejemplo, considere que hay  $n$  tipos de elementos indistinguibles  $n_1 + n_2 + \dots + n_p = n$ .

Considere el siguiente ejemplo sencillo. Se nos pide que encontremos formas de reordenar las letras de la palabra "Mañana". La longitud de la palabra es  $n = 8$ , pero calculando una permutación  $P(n) = n!$  es incorrecto, ya que en permutaciones o r-permutaciones asumimos que todos los elementos son diferentes (para evitar contar en exceso). En otras palabras,  $n!$  cuenta la palabra "Mañana" varias veces:  $\binom{n}{3}$  para las "o" y  $\binom{n}{2}$  para las "r", ya que tenemos  $n_2 = 3$  "r" y  $n_4 = 2$ . Las "o" (el resto de las letras "T", "m" y "w") aparecen solo una:  $n_1 = n_3 = n_5 = 1$ .

Resumiendo, tenemos 5 tipos de letras en la palabra "Mañana" y  $n_1 + n_2 + n_3 + n_4 + n_5 = n$ , es decir  $1 + 3 + 1 + 2 + 1 = 8 = n$ . Entonces, si aplicamos la regla del producto y descomponemos nuestro problema en 5 tareas de contar (tantas como diferentes tipos de letras), cada una de estas tareas es una  $r$ -combinación de la siguiente manera:

- Tenemos  $n = 8$  puestos a cubrir.
- Llenar  $n_2 = 3$  posiciones con  $\binom{n}{n_2=3} = \binom{8}{3}$  combinaciones. Usamos combinaciones ya que una vez seleccionamos cualquiera de los grupos de 3 posiciones, el orden no importa dentro del grupo.
- Ahora tenemos  $n - n_2 = 8 - 3 = 5$  puestos restantes por cubrir. Llenar  $n_4 = 2$  de ellos con  $\binom{n-n_2}{n_4} = \binom{5}{2}$  combinaciones.
- Ahora hay  $n - n_2 - n_4 = 3$  posiciones a la izquierda. Como  $n_1 = n_3 = n_5 = 1$ , tome cualquier orden de ellos para hacer  $\binom{3}{1} \cdot \binom{2}{1} \cdot \binom{1}{1}$  combinaciones.

Finalmente, el producto de todo lo anterior  $r$ -combinaciones conducen a

$$\begin{aligned} \binom{8}{3} \cdot \binom{5}{2} \cdot \binom{3}{1} \cdot \binom{2}{1} \cdot \binom{1}{1} &= \frac{8!}{3!5!} \cdot \frac{5!}{2!3!} \cdot \frac{3!}{1!2!} \cdot \frac{2!}{1!1!} \cdot \frac{1!}{1!0!} \\ &= \frac{8!}{\cancel{3!} \cancel{5!} \cdot \cancel{2!} \cancel{3!} \cdot \cancel{1!} \cancel{2!} \cdot \cancel{1!} \cancel{1!} \cdot \cancel{1!} \cancel{0!}} \\ &= \frac{8!}{3!2!1!1!1!} . \end{aligned}$$

Entonces, **permutaciones con repetición**.  $P_\sigma(n)$  se definen de la siguiente manera:

$$P_\sigma(n) = \frac{n!}{n_1!n_2!\dots n_p!}$$

donde tenemos  $p$  tipos de objetos indistinguibles  $y n_1 + n_2 + \dots + n_p = n$ .

La definición anterior también se puede interpretar de la siguiente manera. En el numerador tenemos *todas* las permutaciones. Sin embargo, en el denominador cancelamos algunos de ellos para evitar contar en exceso.

Esto es consistente con la definición de  $C(n, r)$  como  $P(n, r)/P(r)$ . Por ejemplo, supongamos que  $n_1 \geq n_2 \geq \dots \geq n_p$ .

$$P_\sigma(n) = \frac{P(n, n_1)}{P(n_1)} \cdot \frac{P(n - n_1, n_2)}{P(n_2)} \cdots \frac{P(n - (n_1 + n_2 + \dots + n_{p-1}), n_p)}{P(n_1 + n_2 + \dots + n_{p-1})} .$$

### 2.2.3.2. Conjuntos múltiples

Los problemas de conteo con repeticiones, independientemente de que haya elementos indistinguibles o no, se pueden ver desde el ángulo de **las multiseries**. ¿Por qué? Porque un multiconjunto es una generalización de un conjunto donde se permite que los elementos se repitan (podemos incluso tener infinito, es decir  $\infty$ , copias).

Por ejemplo, las letras de la palabra "Mañana" están codificadas por el siguiente conjunto múltiple:

$$S = \{1 \cdot T, 3 \cdot o, 1 \cdot m, 2 \cdot r, 1 \cdot w\} .$$

dónde  $n_1 = 1, n_2 = 3, n_3 = 1, n_4 = 2, n_5 = 1$  y  $n_1 + n_2 + n_3 + n_4 + n_5 = n = 8$ .

Este nos permite explicar mejor el número de **permutaciones con repeticiones** (en el sentido indistinguible) como producto

[Saltar al contenido principal](#)

$$P_\sigma(n) = \binom{n}{n_1} \cdot \binom{n-n_1}{n_2} \cdot \binom{n-n_1-n_2}{n_3} \cdots \binom{n-n_1-\dots-n_{p-1}}{n_p},$$

ya que tomamos primeras  $n_1$  posiciones con el primer elemento del multiset, luego tomamos  $n_2$  del resto  $n - n_1$  posiciones con el segundo elemento del multiset, etc.

En particular, cuando no se permiten repeticiones, todos los elementos del (multi)conjunto tienen *una sola copia*:  $n_1 = n_2 = \dots = n_p = 1$ . Entonces tenemos la definición de **r-permutaciones**:

$$\begin{aligned} P(n, r) &= n \cdot (n - n_1) \cdot (n - n_1 - n_2) \cdots (n - n_1 - n_2 - \dots - n_r) \\ &= n \cdot (n - 1) \cdot (n - 2) \cdots (n - r). \end{aligned}$$

Entonces, usando ahora

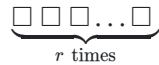
$$S = \{1 \cdot T, 1 \cdot o, 1 \cdot m, 1 \cdot r, 1 \cdot w\},$$

$P(n = 5, r)$  da el número de palabras de longitud  $r \leq n$  que se puede generar usando (una vez) las letras en  $S$ . ¡Por supuesto, "Mañana" nunca aparece!

Además, podemos usar conjuntos múltiples para contar **r-permutaciones con repetición**.  $P_\sigma(n, r) = n^r$  en el entorno distingible, simplemente asumiendo que todos los elementos en el conjunto múltiple **tienen copias infinitas**, por ejemplo, redefinamos  $S$  como

$$S = \{\infty \cdot T, \infty \cdot o, \infty \cdot m, \infty \cdot r, \infty \cdot w\}.$$

Entonces nosotros tenemos  $n = 5$  elementos con infinitas copias cada uno en  $r$  posiciones que pueden ocupar, **ahora llamamos a estas posiciones cajas con capacidad infinita**:



Como tenemos infinitas copias de "T", "o", "m", "r" y "w", tenemos

- $n$  opciones para llenar el primero de los  $r$  posiciones.
- De nuevo, tenemos  $n$  opciones para llenar el segundo de los  $r$  posiciones, etc

Entonces nosotros tenemos

$$P_\sigma(n, r) = \underbrace{n \cdot n \cdot n \cdots n}_{r \text{ times}} = n^r.$$

En este último caso, "Mañana" aparece si  $r = 8$ .

Resumiendo hasta ahora, **las permutaciones con repetición de un conjunto son las permutaciones de un multiconjunto adecuadamente definido!**

### 2.2.3.3. Combinaciones con repetición

Los multisets son especialmente útiles para definir **combinaciones con repetición** como **combinaciones en un multiset**.

En primer lugar, recuerda el significado del número combinatorio.  $\binom{n}{r}$  es el número de subconjuntos de  $r$ -elementos de **cualquier conjunto de  $n$  (diferentes) elementos**.

Consideremos, por ejemplo, el conjunto  $[n] = \{1, 2, \dots, n\}$  (una notación muy útil para usar a partir de ahora). Entonces,  $\binom{n}{2}$  es el numero de  $S_2 \subseteq [n]$  con  $|S_2| = 2$ , es decir  $S_2 = \{(1, 2), (1, 3), \dots, (1, n)\}$ . En general, para  $S_r \subseteq [n]$ ,  $|S_r| = \binom{n}{r}$ . En realidad, sabemos que  $|S_0| + |S_1| + \dots + |S_n| = 2^n$  debido a la biyección entre el conjunto de potencia y  $\{0, 1\}^n$ .

Por lo tanto, en términos de codificación binaria, un conjunto  $S_r$  está dado por los números binarios de  $n$  bits con *exactamente*  $r$  unos ( $n - r$  ceros). Entonces podemos distribuir el  $r$  entre las  $n$  posiciones. En consecuencia, tendremos  $n - r$  ceros distribuidos en bloques de  $x_i$  ceros cada uno, que se colocarán "antes" y "después" de cada  $i$ -posición (indicada por  $1^i$ ) de un 1:

[Saltar al contenido principal](#)

$$\underbrace{\square \square \dots \square}_{x_1 \text{ 0s}} \underbrace{1^1 \square \square \dots \square}_{x_2 \text{ 0s}} \underbrace{1^2 \square \square \dots \square}_{x_3 \text{ 0s}} \underbrace{1^3 \dots \square \square \dots \square}_{x_r \text{ 0s}} \underbrace{1^r \square \square \dots \square}_{x_{r+1} \text{ 0s}}$$

Tenga en cuenta que elegir las  $r$  posiciones para aquellos entre los  $n$ . Las ranuras disponibles determinan la ubicación de los ceros. En otras palabras, las posiciones de los unos indican qué elementos de  $[n]$  pertenece a un subconjunto determinado. Esta es simplemente la definición del número combinatorio.

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} = P_\sigma(n) = \frac{n!}{n_1!n_2!}$$

dónde  $n_1 = r, n_2 = (n - r)$  y  $n_1 + n_2 = n$ . En otras palabras,  $\binom{n}{r}$  es el número de **permutaciones del multiconjunto**

$$S_r = \{r \cdot 1, (n - r) \cdot 0\}$$

Entonces, considere ahora el siguiente conjunto múltiple

$$T = \{\infty \cdot x_1, \infty \cdot x_2, \dots, \infty \cdot x_m\},$$

dónde  $m = r + 1$ . Entonces, el problema de encontrar todos los submulticonjuntos de  $T$  de tamaño  $k$  ( $r \leq k \leq n$ ) se puede plantear en términos de encontrar el número de soluciones enteras no negativas de la ecuación

$$x_1 + x_2 + x_3 + \dots + x_m = k,$$

ya que podemos resolver esta ecuación tomando  $x_i \geq 0$  de cualquiera de los  $m$  tipos, con la restricción de que la suma de elementos de todos los tipos sea *exactamente*  $k$ . Esto lo cuantificamos de la siguiente manera:

- Por lo tanto, en la práctica sólo necesitamos  $x_i \leq k$  (a pesar del número de copias en formalmente  $\infty$ ). De hecho, encontrar un subconjunto múltiple consiste en mezclar elementos de algunos tipos  $x_i$  siempre que la suma de todas las opciones sea  $k$ .
- Para visualizar mejor cada sub-multiconjunto, consideramos  $x_i$  como una **caja de capacidad infinita** (en la práctica su capacidad máxima es  $k$ ). Todas las casillas son diferentes y están separadas por un "+" que marca el cambio de caja  $x_{i-1}$  empacar  $x_i$  para  $i = 2, \dots, m$ , es decir, tenemos  $m - 1$  **separadores**.
- Esto significa que el  $m - 1$  Hay que tener en cuenta los separadores aunque no podemos incluirlos en el multiset.  $T$  ya que no se pueden secuenciar arbitrariamente sino que aparecen entre dos bloques/cuadros  $x_{i-1}$  y  $x_i$ .
- En resumen, tenemos que contar el número de  $k$ - **combinación** de  $k + (m - 1)$  símbolos (eligiendo exactamente  $k$  elementos). Este número se define como

$$\binom{m}{k} = \binom{k+m-1}{k} = \binom{k+m-1}{n-1}.$$

y se lee como " $m$  selección múltiple  $k$ ".

Curiosamente, desde  $m = r + 1$  y  $k = n - r$  tenemos

$$\binom{m}{k} = \binom{k+m-1}{k} = \binom{(n-r)+(r+1)-1}{n-r} = \binom{n}{n-r} = \binom{n}{r},$$

es decir, el número de  $n - r$  números de bits con exactamente  $r$  bits puestos a uno! (dónde  $n = m - 1$  y  $r = n - k$ ).

Las, las **combinaciones con repetición**.  $C_\sigma(m, k)$  de  $k$  elementos de  $m$  tipos son:

$$C_\sigma(m, k) = \binom{k+m-1}{k},$$

desde el punto de vista de la colocación  $k$  elementos del multiset (con repetición), y

[Saltar al contenido principal](#)

$$C_\sigma(m, m-1) = \binom{k+m-1}{m-1},$$

desde el punto de vista de la colocación del  $m-1$  marcadores (también con repetición pero restringidos a elementos separados de diferentes tipos).

**Capacidades mínimas**. Hasta ahora, los elementos del conjunto múltiple tienen capacidad infinita, pero estas capacidades **no tienen límites inferiores**. Sin embargo, ¿qué sucede si todos los elementos del multiconjunto deben satisfacer  $x_i \geq a_i$ ?

Queremos encontrar las soluciones enteras no negativas para la ecuación.

$$x_1 + x_2 + x_3 + \dots + x_m = k, \text{ subject to } x_i \geq a_i \geq 0 \forall i.$$

Evidentemente, este problema también tiene solución si  $a_i \leq k$  para cualquiera  $i$ . Entonces, la estrategia a seguir es:

- *Donar  $a_i$  elementos para  $x_i$  caja.*
- Definir nuevas variables  $y_i = x_i - a_i$
- Dejar  $A = a_1 + a_2 + a_3 + \dots + a_m$ . Entonces, el *nuevo* problema a resolver es encontrar las soluciones enteras no negativas de

$$y_1 + y_2 + y_3 + \dots + y_m = k - A$$

**Ejercicio**. Encuentra el número de formas de dar 6 naranjas a 3 niños para que **todos los niños tengan al menos una** naranja.

- Comenzamos *identificando* qué son las cajas y cuáles los elementos. En este caso, los niños son las cajas y las naranjas son los elementos para llenar las cajas (una caja por niño).
- Entonces, el problema (básico) a resolver es encontrar las soluciones enteras no negativas de  $x_1 + x_2 + x_3 = 6$ . La solución es  $C_\sigma(m=3, k=6) = \binom{6}{3} = \binom{6+3-1}{3} = \binom{8}{3} = 8!/(6! \cdot 2!) = 28$ .

Sin embargo, **esta no es** la solución al problema. Dado que **todos** los niños deben tener al menos una naranja, nos interesan las soluciones donde  $x_i \geq 1 \forall i$ .

Entonces, le **damos** 1 naranja para cada niño (podemos hacer eso porque  $6 \geq 3$ ) y resuelve el siguiente problema:

$$y_1 + y_2 + y_3 = 6 - 1 \cdot 3 = 3, \text{ donde } y_i = x_i - 1.$$

La **solución** es  $C_\sigma(m=3, k=3) = \binom{3}{3} = \binom{3+3-1}{3} = \binom{5}{3} = 5!/(3! \cdot 2!) = 10$ . Estas soluciones (wrt las  $y_i$ s y asignado a  $x_i$  después de deshacer el cambio de variable) son:

$$\begin{aligned} (1\ 1\ 1) &\rightarrow (2\ 2\ 2) \\ (2\ 0\ 1) &\rightarrow (3\ 1\ 2) \\ (2\ 1\ 0) &\rightarrow (3\ 2\ 1) \\ (1\ 0\ 2) &\rightarrow (2\ 1\ 3) \\ (1\ 2\ 0) &\rightarrow (2\ 3\ 1) \\ (0\ 1\ 2) &\rightarrow (1\ 2\ 3) \\ (0\ 2\ 1) &\rightarrow (1\ 3\ 2) \\ (3\ 0\ 0) &\rightarrow (4\ 1\ 1) \\ (0\ 3\ 0) &\rightarrow (1\ 4\ 1) \\ (0\ 0\ 3) &\rightarrow (1\ 1\ 4) \end{aligned}$$

Podemos concluir que  $28 - 10 = 18$ . Las soluciones **no satisfacen** el requisito de que cada niño tenga al menos una naranja. Todas estas soluciones incorrectas ocurren cuando al menos un niño tiene 0 naranjas.

Tenga en cuenta que la segunda parte del ejercicio anterior tendría una **solución diferente** si ningún niño puede tener más de 1 naranja. Esto requiere otra estrategia.

**Capacidades Máximas**. Si todas las cajas satisfacen  $0 \leq x_i \leq a_i$  (su capacidad está **limitada por el límite superior**) tenemos conjuntos múltiples como

[Saltar al contenido principal](#)

$$T = \{a_1 \cdot x_1, a_2 \cdot x_2, \dots, a_m \cdot x_m\},$$

es decir, queremos encontrar las soluciones enteras no negativas para la ecuación

$$x_1 + x_2 + x_3 + \dots + x_m = k, \text{ s.t. } 0 \leq x_i \leq a_i \forall i.$$

Cuenta primero las combinaciones con capacidades infinitas y luego reste las *combinaciones indeseables*.

Las combinaciones indeseables donde  $x_i$  está involucrado satisfacer  $x_i > a_i$ , eso es  $x_i \geq a_i + 1$ .

- Definamos  $A_i$  correspondiente a  $x_i$  como:

$$A_i = \text{Sub-multiset of size } k \text{ with } x_i \geq a_i + 1$$

- Tenemos  $A_1, A_2, \dots, A_m$  multiconjuntos, cada uno correspondiente a una *restricción inversa*  $x_i \geq a_i + 1$  y queremos contar el número de combinaciones que NO SATISFACEN NINGUNA de las restricciones inversas. En otras palabras, esto equivale a encontrar el tamaño del conjunto:

$$\bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_m = \overline{A_1 \cup A_2 \cup \dots \cup A_m}.$$

- Informática | $\bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_m$ | requiere utilizar el **principio de inclusión-exclusión**. Aquí lo definimos y más adelante lo explicamos detalladamente. Este principio simplemente surge de la generalización de la ecuación.

$$|A \cup B| = |A| + |B| - |A \cap B|;$$

$$\begin{aligned} |A_1 \cup A_2 \cup \dots \cup A_m| &= \sum_{i=1}^m |A_i| - \sum_{1 \leq i < j \leq m} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq m} |A_i \cap A_j \cap A_k| \\ &\quad + \dots + (-1)^m |A_1 \cap A_2 \cap \dots \cap A_m|. \end{aligned}$$

La ecuación anterior establece que para contar los elementos de una unión de conjuntos:

- Primero cuenta los elementos de cada conjunto:  $|A_1| + |A_2| + \dots$
- Al hacerlo, si algún par de conjuntos  $|A_i \cap A_j| > 0$  esto significa que algunos elementos están *sobre contados* al hacer  $|A_i| + |A_j|$ . Entonces,  $|A_i \cap A_j|$  debe descontarse de la suma  $\sum_{i=1}^m |A_i|$ .
- A continuación, si algún triplete de conjuntos satisface  $|A_i \cap A_j \cap A_k| > 0$ , dónde  $|A_i \cap A_j| > 0$ , esto significa que descontar esa intersección debe compensarse sumando  $|A_i \cap A_j \cap A_k|$ .
- Luego, la progresión continúa sumando términos impares y restando pares.

Como resultado, dejemos  $S$  un conjunto dado (en nuestro caso, el conjunto de todas las combinaciones del multiconjunto con capacidad infinita) y  $A_i \subseteq S$ . Entonces, el principio de inclusión-exclusión se formula de la siguiente manera:

$$|\bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_m| = |S| - |A_1 \cup A_2 \cup \dots \cup A_m|.$$

Para contar estas combinaciones necesitamos:

- Calcula todos los  $|A_i|$  y sumarlos. En particular, tenemos

$$|A_i| = \binom{m}{k - (a_i + 1)}.$$

- Calcula todos los  $|A_i \cap A_j|$  y descontarlos de la suma de  $|A_i|$
- Y así sucesivamente con  $|A_i \cap A_j \cap A_k|$ , etc.

Hagámoslo en el siguiente ejercicio.

**Ejercicio**. Disponemos de varios tipos de bolas: 4 de ellos son rojos (R), 3 son verdes (G) y 2 son azules (B). Encuentra el número de formas de obtener 8 pelotas. Tenemos el multiset  $T = \{4 \cdot R, 3 \cdot G, 2 \cdot B\}$ , es decir  $m = 3$  (tipos de bolas) y  $k = 8$  bolas para tomar en total.

[Saltar al contenido principal](#)

Tenemos que encontrar todas las soluciones enteras no negativas de:

$$x_1 + x_2 + x_3 = 8 \text{ s.t. } 0 \leq x_1 \leq 4, 0 \leq x_2 \leq 3, 0 \leq x_3 \leq 2.$$

Definamos los submulticonjuntos del multiconjunto con capacidad infinita  $T^* = \{\infty \cdot R, \infty \cdot G, \infty \cdot B\}$  para las tres restricciones:

$A_1$  = Sub-multiset of size 8 with  $x_1 \geq 5$

$A_2$  = Sub-multiset of size 8 with  $x_2 \geq 4$

$A_3$  = Sub-multiset of size 8 with  $x_3 \geq 3$ .

Dejar  $S$  ser el conjunto de todos los sub-multiconjuntos de tamaño 8 de  $T^*$ . Su tamaño es el multi-elección.

$$|S| = \binom{m}{k} = \binom{3}{8} = \binom{10}{8} = 45.$$

¿Cuál es el tamaño de cada uno?  $A_i$ ? Bueno, por definición están dados por

$$|A_1| = \binom{m}{k - (a_1 + 1)} = \binom{3}{8 - 5} = \binom{3}{3} = \binom{5}{3} = 10.$$

$$|A_2| = \binom{m}{k - (a_2 + 1)} = \binom{3}{8 - 4} = \binom{3}{4} = \binom{6}{4} = 15.$$

$$|A_3| = \binom{m}{k - (a_3 + 1)} = \binom{3}{8 - 3} = \binom{3}{5} = \binom{7}{5} = 21.$$

Sin embargo, encontrar los tamaños de las intersecciones binarias  $|A_i \cap A_j|$  puede que esto no sea tan claro. Para  $A_1 \cap A_2$  es porque debemos satisfacer ambos  $x_1 \geq 5$  y  $x_2 \geq 4$  pero esto excede nuestro presupuesto que es  $k = 8$  y luego  $A_1 \cap A_2 = \emptyset$ . Entonces  $|A_1 \cap A_2| = 0$

Si embargo, para  $A_1 \cap A_3$  tenemos que las restricciones son  $x_1 \geq 5$  y  $x_3 \geq 3$  y la suma de los límites superiores  $5 + 3 = 8$  es igual al presupuesto. Por lo tanto, dando 5 + 3 bolas para  $x_1$  y  $x_3$  efectivamente  $x_2 = 0$  y tenemos una combinación única ( $x_1 = 5, x_2 = 0, x_3 = 3$ ). Por lo tanto  $|A_1 \cap A_3| = 1$

$A_2 \cap A_3$  debe satisfacer  $x_2 \geq 4$  y  $x_3 \geq 3$  y tenemos  $4 + 3 = 7$ . Después de configurar  $x_2 = 4$  y  $x_3 = 3$ , tenemos soluciones del tipo  $(r_1, 4 + r_2, 3 + r_3)$  que podemos encontrar configurando  $y_2 = 4 - x_2, y_3 = 3 - x_3$  y encontrar las soluciones de  $x_1 + y_2 + y_3 = 1$ . El número de soluciones es  $|A_2 \cap A_3| = \binom{m}{1} = \binom{3}{1} = \binom{3}{1} = 3$ .

Finalmente,  $A_1 \cap A_2 \cap A_3 = \emptyset$  y  $|A_1 \cap A_2 \cap A_3| = 0$ .

Resumiendo, tenemos

$$\begin{aligned} |\bar{A}_1 \cap \bar{A}_2 \cap \bar{A}_3| &= |S| - (|A_1| + |A_2| + |A_3|) + (|A_1 \cap A_2| + |A_1 \cap A_3| + |A_2 \cap A_3|) - |A_1 \cap A_2 \cap A_3| \\ &= 45 - (10 + 15 + 21) + (0 + 1 + 3) - 0 \\ &= 45 - 46 + 4 - 0 = 3. \end{aligned}$$

Estas soluciones son: (4, 3, 1), (4, 2, 2) y (3, 3, 2). Sólo estas soluciones satisfacen

$$x_1 + x_2 + x_3 = 8 \text{ s.t. } 0 \leq x_1 \leq 4, 0 \leq x_2 \leq 3, 0 \leq x_3 \leq 2.$$

es decir, la condición  $x_1 + x_2 + x_3 = 8$  nos obliga a informar asignaciones cercanas a los límites superiores.

## 2.3. El Principio de Exclusión-Inclusión

El **Principio de Exclusión-Inclusión (PEI)** es una herramienta poderosa para contar problemas con restricciones, es decir, cuando tenemos que contar en exceso (incluir) y luego eliminar (excluir) conteos. El PIE es muy útil para nuestro Cerebro Discreto (ver el problema antes mencionado de contar combinaciones con repeticiones bajo restricciones). A menudo se considera una forma "ingenua" de resolver problemas de conteo con restricciones porque su complejidad es  $O(2^n)$  pero es muy útil para comprender la naturaleza combinatoria de un problema como [contar caminos en una cuadrícula con posiciones prohibidas](#), como descubriremos al final de esta sección.

[Saltar al contenido principal](#)

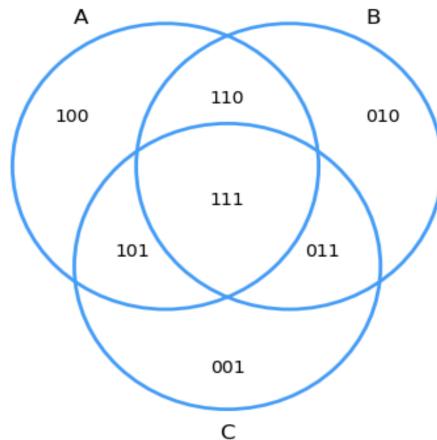


Fig. 2.3 Codificación binaria de todos los subconjuntos.

Considere el escenario general en **PEI1**, donde tenemos  $m = 3$  conjuntos que se cruzan:  $A_1 = A, A_2 = B, A_3 = C$ . Comenzamos eligiendo una *codificación útil* para cada uno de los distintos subconjuntos. Por ejemplo, 100 significa el subconjunto  $A_1 \cap \bar{A}_2 \cap \bar{A}_3$  (elementos que sólo pertenecen a  $A_1$ ), donde 101 medio  $A_1 \cap \bar{A}_2 \cap A_3$ .

Curiosamente, la codificación binaria antes mencionada nos dice cuántas veces se **sobrecuentan** los elementos de un subconjunto determinado (solo tome los números de unos en el código). Esto será muy útil a partir de ahora.

### 2.3.1. Descuento de conteos excesivos

Ahora comencemos contando los elementos de la unión.  $\bigcup_{i=1}^m A_i$  demconjuntos  $A_1, A_2, \dots, A_m$  simplemente sumando los elementos de los conjuntos individuales:

$$\bigcup_{i=1}^m A_i = \sum_{i=1}^m |A_i| .$$

Según **PEI1**, los códigos binarios de cada subconjunto indican cuántas veces se cuentan los elementos de ese subconjunto. Esto resulta en

$$A_1 = A = \{100, 110, 101, 111\}, A_2 = B = \{010, 011, 110, 111\}, A_3 = C = \{001, 101, 111, 011\}$$

es decir, hasta ahora nuestra unión puede verse como el siguiente conjunto múltiple:

$$A_1 \cup A_2 \cup A_3 = \{100, 010, 001, 2 \cdot 110, 2 \cdot 101, 2 \cdot 011, 3 \cdot 111\}$$

Cada  $A_i$  tiene elementos contados una vez (1de ellos), dos veces (2de ellos) y tres veces (1de ellos).

- Se debe eliminar uno de los elementos contados dos veces.
- Se deben eliminar dos de los elementos contados tres veces.

Para eliminar los elementos que se cuentan dos veces, eliminamos los elementos en todas las intersecciones por pares:  $A \cap B = \{110, 111\}, A \cap C = \{101, 111\}$  y  $B \cap C = \{011, 111\}$  como se indica en **PEIPairwise**. Tenga en cuenta que el subconjunto 111 se elimina tres veces.

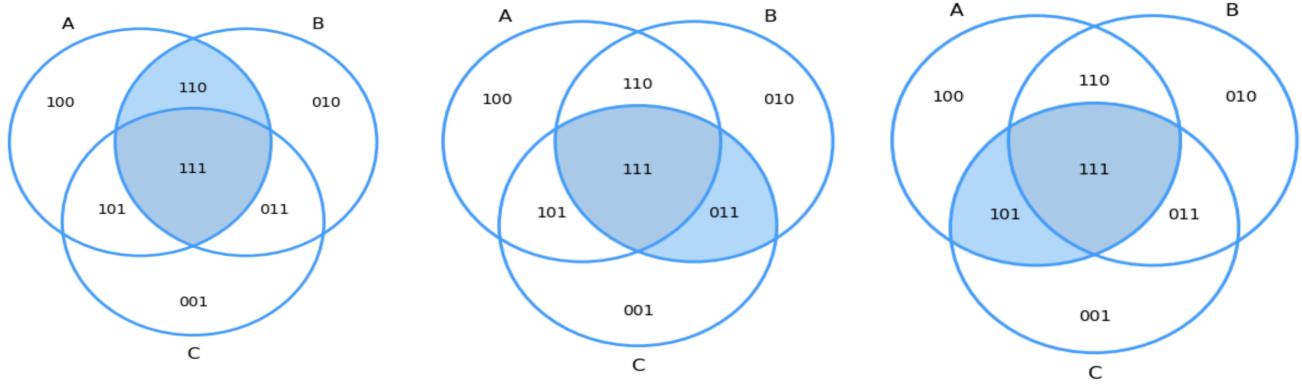


Fig. 2.4 Por pares excluyendo intersecciones.

De manera más general, hacemos

$$\bigcup_{i=1}^m A_i = \sum_{i=1}^m |A_i| - \sum_{1 \leq i < j \leq m} |A_i \cap A_j| .$$

Después de eliminar el número de elementos en las intersecciones por pares, la unión resultante es

$A_1 \cup A_2 \cup A_3 = \{100, 010, 001, 110, 101, 011\}$ . es decir, las tres copias de 111 son removidas. Es por esto que se puede incluir un conteo correcto  $A_1 \cap A_2 \cap A_3 = \{111\}$ .

De manera más general, para  $m = 3$  tenemos:

$$\bigcup_{i=1}^m A_i = \sum_{i=1}^m |A_i| - \sum_{1 \leq i < j \leq m} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq m} |A_i \cap A_j \cap A_k| .$$

Por un arbitrario  $m$  obtenemos:

$$\begin{aligned} |A_1 \cup A_2 \cup \dots \cup A_m| &= \sum_{i=1}^m |A_i| - \sum_{1 \leq i < j \leq m} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq m} |A_i \cap A_j \cap A_k| \\ &\quad + \dots + (-1)^m |A_1 \cap A_2 \cap \dots \cap A_m| . \end{aligned}$$

Podemos formular la fórmula anterior de manera más compacta. Para hacer eso, tenga en cuenta que el tamaño de la intersección aumenta linealmente ( $1, 2, 3, \dots, m$ )

$$\bigcup_{i=1}^m A_i = \sum_{k=1}^m (-1)^{k+1} \left( \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq m} |A_{i_1} \cap A_{i_2} \dots \cap A_{i_k}| \right) .$$

Podemos ser aún más compactos al darnos cuenta de que el  $k$ -ésimo término de la suma principal se basa en sumar  $\binom{m}{k}$  intersecciones.

Esto significa que calcular el tamaño de la unión es equivalente a calcular todos los subconjuntos posibles de  $[m] = \{1, 2, \dots, m\}$  pero el conjunto vacío y los interseca de la siguiente manera:

$$\bigcup_{i=1}^m A_i = \sum_{\emptyset \neq J \subseteq [m]} (-1)^{|J|+1} \bigcap_{j \in J} A_j .$$

En consecuencia, dado que tenemos que visitar todos los subconjuntos de  $[m]$ , esta forma *ingenua* de calcular el tamaño de la unión se puede hacer en  $O(2^m)$ .

[Saltar al contenido principal](#)

### 2.3.2. Principio de Exclusión-Inclusión

Una vez que hemos contado el tamaño de la unión de conjuntos, el **PEI** es algo natural: el número de elementos que NO PERTENECEN a ninguno de los conjuntos  $A_1, A_2, \dots, A_m$  (es decir, pertenecen a  $\bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_m$ ) es dado por:

$$\bigcap_{i=1}^m \bar{A}_i = |S| - \bigcup_{i=1}^m \bar{A}_i ,$$

dónde  $S$  es un conjunto que incluye todos los  $A_i$  (el **conjunto universal**). En nuestro ejemplo anterior de **PEI**, si  $S = \bigcup_{i=1}^m A_i$ , entonces la intersección de complementos  $\bar{A}_1 \cap \bar{A}_2 \cap \dots \cap \bar{A}_m$  puede ser codificado por 000 y esto significa que no tiene elementos dentro de ningún diagrama de Venn!

### 2.3.3. Contando caminos en grillas

Una de las aplicaciones más interesantes del PEI en IA es **el conteo de rutas**. Dada una **cuadrícula** (más adelante la definiremos de manera más formal), nos interesa contar (y también encontrar) los caminos entre dos **nodos** de esta cuadrícula, digamos  $(0, 0)$  y  $(m, n)$ .

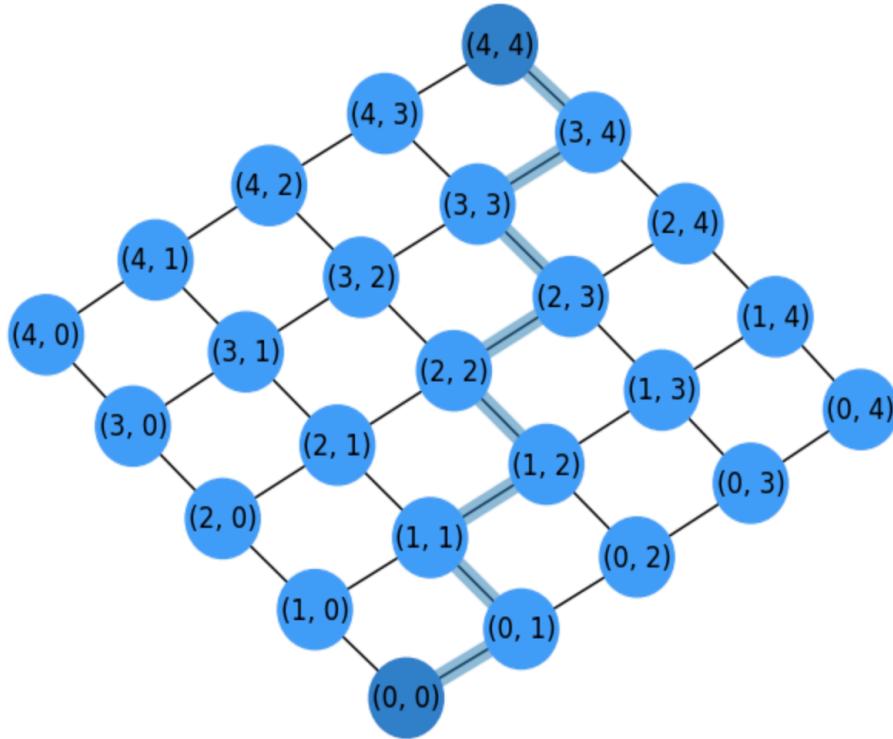


Fig. 2.5 Cuadrícula para contar caminos entre  $(0, 0)$  y  $(4, 4)$ .

En **grid1** mostramos una grilla donde  $m = n = 4$ . Un **nodo** dado  $(i, j)$  puede estar conectado como máximo con 4 nodos:  $(i + 1, j)$ ,  $(i, j + 1)$ ,  $(i - 1, j)$  y  $(i, j - 1)$ . Por eso se llama **cuadrícula de 4** porque cada nodo puede tener como máximo 4 **vecinos**.

Sin embargo, para contar caminos entre el origen  $(0, 0)$  y el destino  $(m = 4, n = 4)$  vamos a asumir que estos enlaces están dirigidos, es decir, un nodo  $(i, j)$  sólo puede viajar a  $(i + 1, j)$  o para  $(i, j + 1)$ .

Un **sendero**  $\Gamma$  es solo una secuencia (lista) de nodos entre el origen y el destino. Por ejemplo, la ruta que se muestra en **grid1** es:

$$\Gamma = \{(0, 0), (0, 1), (1, 1), (1, 2), (2, 2), (2, 3), (3, 3), (3, 4), (4, 4)\}$$

Curiosamente, para dibujar un camino en la cuadrícula de arriba solo podemos realizar dos **tipos de movimientos** desde un nodo determinado  $(i, j)$ :

- $\Delta i: (i, j) \rightarrow (i + 1, j)$ .

[Saltar al contenido principal](#)

Entonces: ¿Cuántos caminos existen entre  $(0, 0)$  y  $(m, n)$ ?

Ya que tenemos que llegar a  $(m, n)$  de  $(0, 0)$  necesitamos hacer  $m$  movimientos de tipo  $\Delta i$  y  $n$  movimientos de tipo  $\Delta j$  en un orden determinado!

Como el orden importa, tenemos que encontrar permutaciones de  $m + n$  elementos, donde  $m$  de ellos son de tipo  $\Delta i$  y  $n$  de ellos son de tipo  $\Delta j$ . Tenemos **permutaciones con repetición**:

$$P_\sigma(m+n) = \frac{(m+n)!}{m!n!} = \binom{m+n}{m} = \binom{m+n}{n},$$

es decir, podemos interpretar el conteo en términos de combinaciones, ya que tenemos  $\binom{m+n}{m}$  formas de tomar  $m$  elementos de tipo  $\Delta i$  idem  $n$  elementos mixtos, lo que equivale a la  $\binom{m+n}{n}$  formas de tomar  $n$  elementos de tipo  $\Delta j$ .

Sin embargo, ¿cómo es aplicable el PEI aquí?

Bueno, usemos para eliminar un nodo de la cuadrícula anterior, por ejemplo el nodo  $(2, 2)$ . Como resultado de la eliminación  $(2, 2)$ , el documento mostrado **grid1** no se puede construir. En realidad, la **topología** de la cuadrícula (la forma en que se conectan sus elementos independientemente de su **diseño**, es decir, la forma en que los dibujamos) cambia.

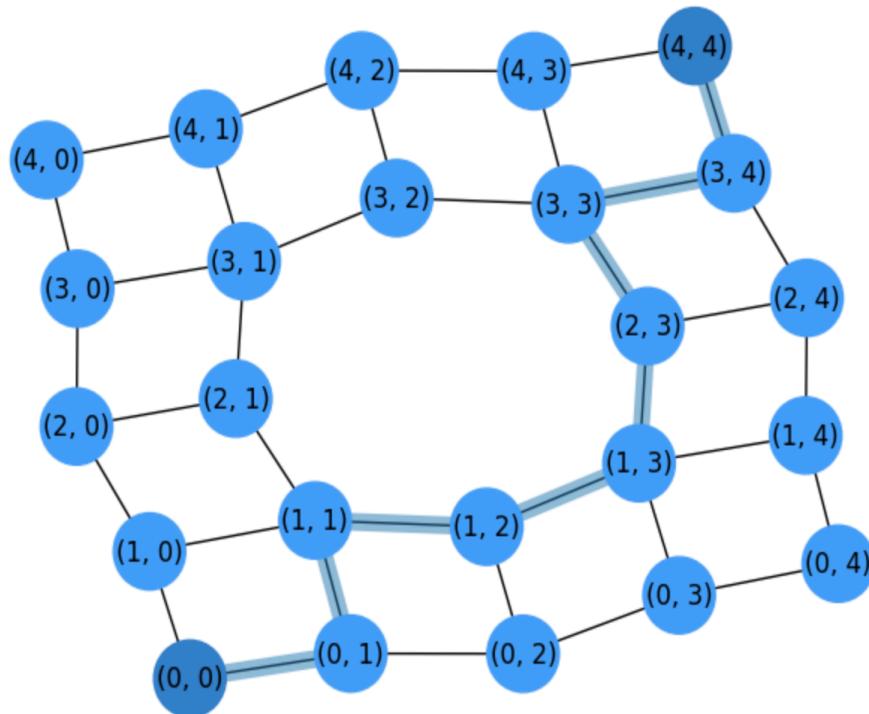


Fig. 2.6 Cuadrícula para contar caminos entre  $(0, 0)$  y  $(4, 4)$  después de quitar el nodo  $(2, 2)$ .

Por ejemplo, al representar la cuadrícula después de eliminar el nodo  $(2, 2)$  Tenemos que sortear este nodo, a través de uno de sus vecinos, como mostramos en **grid2**.

Entonces surge la siguiente pregunta: ¿ Cuántos caminos tenemos después de eliminar un nodo determinado?

Bueno, deja  $N_{ij} = \binom{i+j}{i}$  ser el número de caminos que hacen  $i$  movimientos de tipo  $\Delta i$  y  $j$  movimientos de tipo  $\Delta j$ .

Luego, para encontrar el número de rutas en la cuadrícula de **grid2**, debemos descontar todas las rutas que visitan el nodo eliminado.  $(2, 2)$  del **conjunto universal**  $S_\Gamma$  de todos los caminos entre  $(0, 0)$  y  $(4, 4)$ , cuyo tamaño es  $N_{mn} = N_{44} = \binom{4+4}{4} = 70$ .

El número de caminos que visitan  $(2, 2)$  viene dada por el **producto** de

- El número de caminos que van desde  $(0, 0)$  a  $(2, 2)$ :  $N_{22}$ .
- El número de caminos que van desde  $(2, 2)$  a  $(2, 2)$ :  $N_{(4-2)(4-2)} = N_{22}$ .

[Saltar al contenido principal](#)

$$N_{22} \cdot N_{22} = \binom{4}{2} \cdot \binom{4}{2} = \left( \frac{4!}{2!2!} \right) = 6^2 = 36 .$$

Es un buen ejercicio para ubicarse en **grid1** el 6 caminos que van desde  $(0, 0)$  a  $(2, 2)$  usando 2  $\Delta i$ -se mueve y 2  $\Delta j$ -se mueve. Uno de estos caminos es

$$\Gamma = \{(0, 0), (0, 1), (1, 1), (1, 2), (2, 2)\} .$$

Por lo tanto tenemos que el número de caminos entre  $(0, 0)$  y  $(4, 4)$  en **grid2** es

$$\#\Gamma = N_{44} - N_{22}^2 = 70 - 36 = 34 .$$

Esta es una aplicación ingenua del PEI, donde  $|S| = N_{44}$ .

Sin embargo, eliminemos un segundo nodo, digamos  $(0, 3)$ . La nueva grilla está en **grid2**:

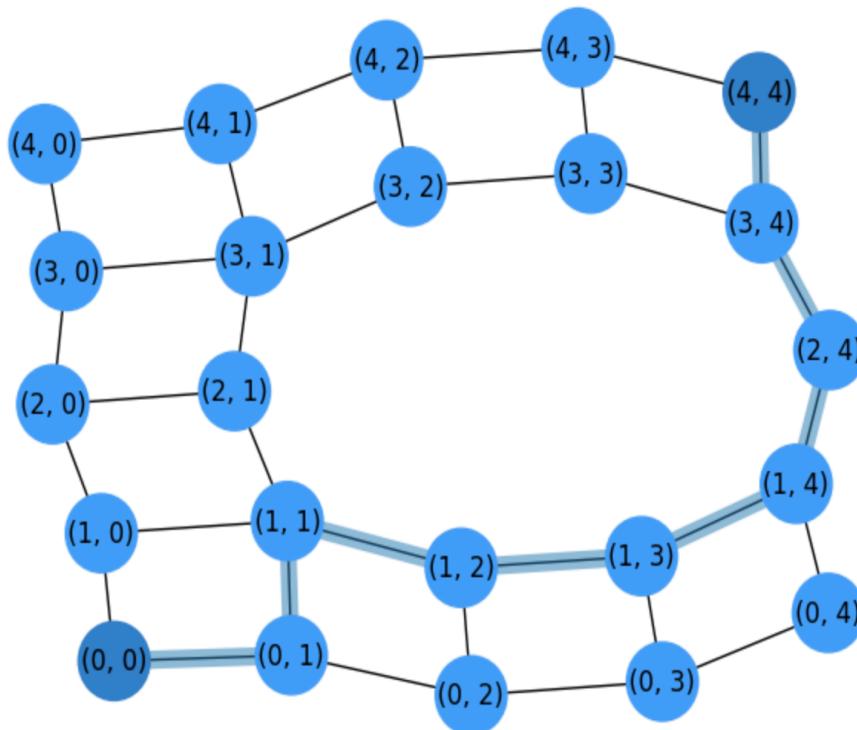


Fig. 2.7 Cuadrícula para contar caminos entre  $(0, 0)$  y  $(4, 4)$  después de eliminar los nodos  $(2, 2)$  y  $(2, 3)$ .

Como podemos ver, el camino resaltado que pasa por  $(2, 3)$  ha sido redirigido y desde  $(1, 3)$  saltamos a  $(1, 4)$ .

Entonces, ¿cuál es el número de caminos después de eliminar ambos?  $(2, 2)$  y  $(2, 3)$ ?

Aplicar el PEI param = 2 conjuntos que debemos considerar:

- $A_1$ : rutas que visitan el nodo  $(2, 2)$ .
- $A_2$ : rutas que visitan el nodo  $(2, 3)$ .

Por tanto, nos interesa el número:

$$\#\Gamma = N_{44} - (|A_1| + |A_2| - |A_1 \cap A_2|) .$$

calculamos  $|A_2|$  como calculamos  $|A_1|$ .

El número de caminos que visitan  $(2, 3)$  viene dada por el **producto** de

- El número de caminos que van desde  $(0, 0)$  a  $(2, 3)$ :  $N_{23}$

[Saltar al contenido principal](#)

Entonces nosotros tenemos

$$|A_2| = N_{23} \cdot N_{21} = \binom{5}{3} \cdot \binom{3}{1} = \left(\frac{5!}{3!2!}\right) \left(\frac{3!}{1!2!}\right) = 10 \cdot 3 = 30 .$$

Finalmente: **Cómo contar los caminos en  $|A_1 \cap A_2|$ ?**

El número de caminos que pasan por ambos lados  $(2, 2)$  y  $(2, 3)$  se puede calcular por el producto de los caminos que van desde  $(0, 0)$  a  $(2, 2)$ , luego pasando de  $(2, 2)$  a  $(2, 3)$  y (finalmente) llegar  $(4, 4)$  desde allí. Estos son:

$$|A_1 \cap A_2| = N_{22} \cdot N_{(2-2)(3-2)} \cdot N_{(4-2)(3-2)} = N_{22} \cdot N_{01} \cdot N_{21} = 6 \cdot 1 \cdot 3 = 18 .$$

Finalmente, tenemos:

$$\begin{aligned} \#\Gamma &= N_{44} - (|A_1| + |A_2| - |A_1 \cap A_2|) \\ &= N_{44} - (N_{22} + N_{23} \cdot N_{21} - N_{22} \cdot N_{01} \cdot N_{21}) \\ &= 70 - (36 + 30 - 18) = 70 - 48 = 22 . \end{aligned}$$

La eliminación de nodos se puede interpretar en términos de marcarlos como nodos *prohibidos* (u *obstáculos*) al planificar el movimiento del robot) como en [grid1withobs](#)

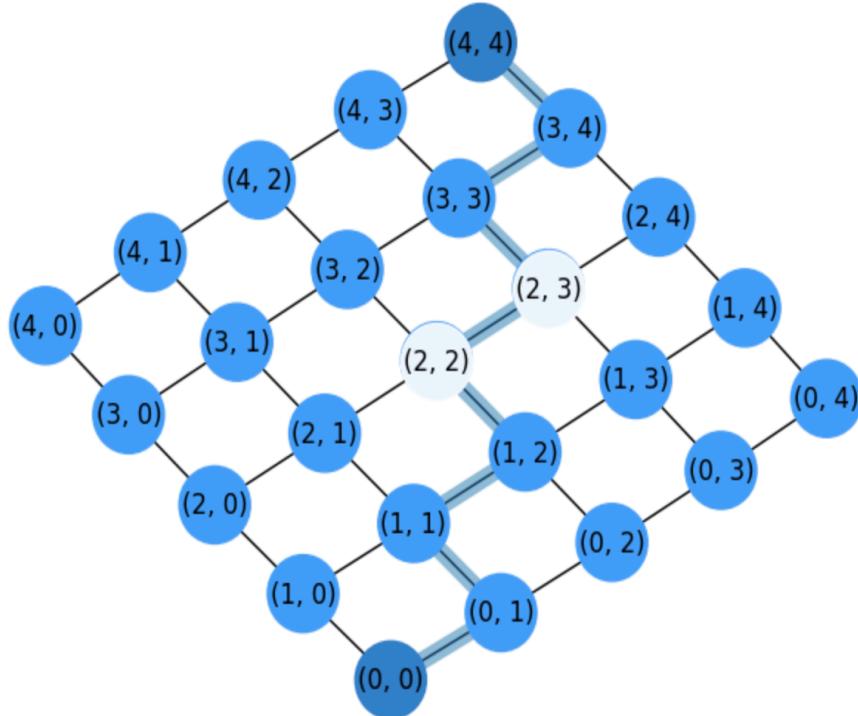


Fig. 2.8 Cuadrícula para contar caminos entre  $(0, 0)$  y  $(4, 4)$  después de marcar los nodos  $(2, 2)$  y  $(2, 3)$  como prohibidos u *obstáculos*.

En general, a medida que eliminamos más y más nodos, aumenta y calcular las intersecciones puede volverse más complejo. Más adelante describiremos formas más sencillas de contar estos caminos y extraerlos.

**Ejercicio** . ¿Qué sucede [grid1withobs](#) si declaramos ese nodo?  $(3, 1)$  ¿Es también un *obstáculo*?

Sólo nos queda descontar los caminos de  $A_3$  (que pasa a través  $(3, 1)$ ) desde  $A_1 \cap A_3 = A_2 \cap A_3 = A_1 \cap A_2 \cap A_3 = \emptyset$ .

Tenemos:  $|A_3| = N_{31} \cdot N_{(4-3)(4-1)} = N_{31} \cdot N_{13} = \binom{4}{1} \cdot \binom{4}{3} = \frac{4!}{1!3!} \cdot \frac{4!}{3!1!} = 4 \cdot 4 = 16 .$

Por lo tanto, tenemos

[Saltar al contenido principal](#)

$$\begin{aligned}\#\Gamma &= N_{44} - (|A_1| + |A_2| + |A_3| - |A_1 \cap A_2|) \\ &= N_{44} - (N_{22} + N_{23} \cdot N_{21} + N_{31} \cdot N_{13} - N_{22} \cdot N_{01} \cdot N_{21}) \\ &= 70 - (36 + 30 + 16 - 18) = 70 - 64 = 6.\end{aligned}$$

Es un buen ejercicio adicional para identificar por qué hacer  $(3, 1)$  un obstáculo reduce la cuenta de caminos desde 22 a solo 6. Básicamente,  $(3, 1)$  reduce a 2 el número de caminos que fluyen por la derecha de la figura, ya que está en la misma fila de  $(2, 2)$ .

## 2.4. La fórmula de Stirling

Informática no puede ser imposible de hacer en una computadora común. Hay limitaciones de memoria porque necesitamos almacenar un producto de  $n$  números y esto puede provocar un desbordamiento de la pila. También existen limitaciones temporales: incluso en el caso de tener memoria infinita, no es posible calcular dos números de longitud arbitraria.  $O(1)$ . Si tiene longitud  $x$ , se necesita  $O(x^{\log_2 3})$  es decir  $O(x^{1.58})$  utilizando el algoritmo de Karatsuba divide y vencerás (ver artículo de Wikipedia [Karatsuba algorithm](#)). Curiosamente,  $x = \text{len}(n)$  crece con  $O(\log_B n)$ , donde  $B$  es la base de  $x$ . Como resultado, calcular el factorial  $n!$  acepta  $O(n \log n^{1.58})$ .

Stirling demostró que si  $n \rightarrow \infty$ ,  $n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$ .

Se puede pensar que la fórmula anterior no reduce la complejidad temporal del factorial ya que implica una  $n^n$  producto y obviamente  $n! < n^n$ . Sin embargo, para la informática  $n$  (o en general  $a^n$ , es decir, una potencia con exponente natural) aprovechamos que todos los factores son iguales (la base) y podemos jugar con los exponentes. En realidad, sólo necesitamos realizar  $O(\log_2 n)$ , en lugar de  $n$ , productos a calcular  $n^n$ .

Prueba. Una forma elegante de probar la afirmación anterior es estimar el número mínimo de productos necesarios para calcular  $a^n$  para  $n \in \mathbb{N}$ . Este método suele consistir en **exponenciación rápida** o **exponenciación al cuadrado**. La intuición es la siguiente. Al factorizar  $a^n$  en el menor número de términos posibles: (1) si el término tiene una potencia par,  $a^{2k}$ , tome valores crecientes de  $k = 2, 4, 8, \dots$  y sigue igual  $k$  tan lejos como sea posible; (2) si finalmente necesita un término con una potencia impar, tome simplemente  $a$ . Por ejemplo, para  $n = 9$  hace  $a^9 = a^4 \cdot a^4 \cdot a$ .

El razonamiento anterior conduce naturalmente a la siguiente relación de recurrencia:

$$T(a, n) = \begin{cases} T(a^2, n/2) & \text{if } n \text{ is even} \\ a \cdot T(a^2, (n-1)/2) & \text{if } n \text{ is odd} \\ 1 & \text{if } n = 0 \end{cases}$$

Consideremos el caso básico donde  $n = 2^k$  es un poder de 2, es decir  $n$  es parejo en cada llamada a  $T$  pero en el último  $n_i = 1$ :

$$\begin{aligned}T(a, n) &= T(a^2, n/2) \\ &= T(a^4, n/4) \\ &= T(a^8, n/8) \\ &= T(a^{2^{k-1}}, n/2^{k-1}) \\ &= T(a^{2^k}, 1) \\ &= a^{2^k} T(a^{2^k} \cdot a^{2^k}, 0) \\ &= a^{2^k} \cdot 1 \\ &= a^{2^k}\end{aligned}$$

Por ejemplo, para  $n = 128 = 2^7$  tenemos

[Saltar al contenido principal](#)

$$\begin{aligned}
T(a, 128) &= T(a^2, 64) \\
&= T(a^4, 32) \\
&= T(a^8, 16) \\
&= T(a^{16}, 8) \\
&= T(a^{32}, 4) \\
&= T(a^{64}, 2) \\
&= T(a^{128}, 1) \\
&= a^{128} T(a^{128} \cdot a^{128}, 0) \\
&= a^{128}.
\end{aligned}$$

Tenga en cuenta que para  $n = 2^k$  Hacemos  $\log_2(n) + 1 = k + 1$  llamadas. El  $k$  las multiplicaciones (descartando la última) son las siguientes

$$\begin{aligned}
a \cdot a &= a^2 \\
a^2 \cdot a^2 &= a^4 \\
a^4 \cdot a^4 &= a^8 \\
a^8 \cdot a^8 &= a^{16} \\
a^{16} \cdot a^{16} &= a^{32} \\
a^{32} \cdot a^{32} &= a^{64} \\
a^{64} \cdot a^{64} &= a^{128}.
\end{aligned}$$

Qué pasa si  $n$  no es una potencia de 2, digamos  $n = 100$ ?

$$\begin{aligned}
T(a, 100) &= T(a^2, 50) \\
&= T(a^4, 25) \\
&= a^4 \cdot T(a^8, 12) \\
&= a^4 \cdot T(a^{16}, 6) \\
&= a^4 \cdot T(a^{32}, 3) \\
&= a^4 \cdot a^{32} T(a^{64}, 1) \\
&= a^4 \cdot a^{32} \cdot a^{64} \cdot T(a^{64} \cdot a^{64}, 0) \\
&= a^4 \cdot a^{32} \cdot a^{64} \cdot 1 \\
&= a^{100}.
\end{aligned}$$

Tenemos  $k = 7$  llamadas y  $\log_2(2^{k-1}) \leq \log_2(100) \leq \log_2(2^k)$ . Los productos son

$$\begin{aligned}
a \cdot a &= a^2 \\
a^2 \cdot a^2 &= a^4 \\
a^4 \cdot (a^4 \cdot a^4) &= a^{12} \\
a^4 \cdot (a^8 \cdot a^8) &= a^{20} \\
a^4 \cdot a^{32} \cdot a^{64} &= a^{100}.
\end{aligned}$$

Tenga en cuenta que agregamos un producto adicional cuando la recurrencia alcanza un valor impar.  $n_i$  en el  $i$ -a llamada. En general tenemos  $O(\log_2 n)$  productos.  $\square$

**Usando la fórmula de Stirling.** Volviendo a la aproximación de Stirling del factorial,

$$n! \sim \frac{n^n}{e^n} \sqrt{2\pi n}$$

Tenemos ambos  $n^n$  (numerador) y  $e^n$  (denominador) se puede calcular de manera eficiente. Números combinatorios igualmente grandes como  $\binom{2n}{n}$  para grandes también se puede calcular de manera eficiente (inténtelo). En general la aproximación de  $n!$  es muy preciso incluso para pequeños  $n$ .

Sin embargo, el interés de la fórmula de Stirling va más allá de las consideraciones de eficiencia:

[Saltar al contenido principal](#)

- *Simplicidad* . La fórmula se compone de funciones simples y esto puede permitirnos diseñar/probar límites para ciertos números combinatorios. Más adelante mostraremos esto junto con el teorema del binomio de Newton.
- *Análisis de complejidad* . Ahora que  $O(n^n) > O(n!)$ , pero ¿qué crece más rápido?  $e^n$  o  $n!$ ? La fórmula de Stirling, o más precisamente su logaritmo, nos permite responder a esta pregunta:

$$\begin{aligned}\log(n!) &\sim \log\left(\frac{n^n}{e^n}\sqrt{2\pi n}\right) = \log(n^n) - \log(e^n) + \frac{1}{2}\log(n) + \frac{1}{2}\log(2\pi) \\ &= n\log n - n + \frac{1}{2}\log n + \frac{1}{2}\log(2\pi).\end{aligned}$$

En este punto es muy útil considerar que para  $n \geq 2$  tenemos

$$n\log n - n < \log(n!) < n\log n,$$

y esto lleva a

$$\log(n!) \sim n\log n.$$

Finalmente, dado que el log es una función monótona de su argumento, consideremos ahora la razón de los logaritmos

$$\frac{\log(n!)}{\log(e^n)} = \frac{n\log n}{n} = \log(n),$$

lo cual es positivo para  $n > 1$ . Por lo tanto,  $O(n!) > O(e^n)$ .  $\square$

- *Límites de información* . Es bien sabido que el número mínimo de comparaciones para ordenar  $n$  números usando un algoritmo de clasificación de comparación es  $\lceil \log(n!) \rceil$ . La intuición viene dada por el algoritmo mergesort: tomar recursivamente 2 sublistas de  $n_i/2$  elementos y cuando las listas tienen un solo elemento devuelven la fusión de listas adyacentes. El número de comparaciones para fusionar dos listas no es mayor que  $n_i$ , y tenemos  $\log_2 n$  niveles. En consecuencia, realizamos al menos  $n \log_2 n$  comparaciones, es decir, tenemos  $\Omega(n \log_2 n) \approx \Omega(\log(n!))$ .
- *Teoría de la información* , en particular la definición combinatoria de entropía.

< Anterior  
[1. El proyecto](#)

Próximo >  
[3. Probabilidad](#)

# Probabilidad

[Imprimir en PDF](#)

## Contenido

- 3.1. Ensayos independientes
- 3.2. Dependencia estadística

Pierre Simon (Marqués de Laplace) en [Un ensayo filosófico sobre probabilidades](#) comenta:

*Los acontecimientos presentes están conectados con los anteriores por un vínculo basado en el principio evidente de que una cosa no puede ocurrir sin una causa que la produzca.*

Esto nos lleva directamente al "casualismo", es decir, estamos hablando de la "probabilidad" de un evento *en el futuro* basado en el conocimiento actual (citando también [las Conferencias de Física Feynmann](#) ).

En realidad, **la probabilidad puede describirse como** la cuantificación del azar . En **probabilidad discreta** hablamos de un conjunto  $\Omega$  (el **conjunto de muestra** ) que contiene todos los **eventos atómicos** posibles . Algunos ejemplos:

$\Omega_1 = \{H, T\}$	Results of tossing a coin: Head (H), Tail (T)
$\Omega_2 = \{1, 2, \dots, 6\}$	Results of playing a dice of 6 faces
$\Omega_3 = \{\omega_1, \dots, \omega_{52!}\}$	Ways of shuffling a standard deck of 52 cards
$\Omega_4 = \{\omega_1, \dots, \omega_{N_{mn}}\}$	Number of paths in a grid

Tenga en cuenta que a veces podemos *nombrar explícitamente* los eventos atómicos, pero otras veces solo podemos *enumerar* cuántos de ellos tenemos. De todos modos, en probabilidad discreta siempre estamos **jugando con conteos** .

En realidad, la probabilidad de un evento particular es la relación entre dos conteos:

- El número de casos **favorables** (a ese evento).
- El número de **todos los casos** .

Para eventos atómicos, su probabilidad es simplemente  $1/|\Omega|$ . Sin embargo, un evento es cualquier proposición que puede evaluarse como verdadera o falsa con una cierta probabilidad, como por ejemplo: "jugar un dado arroja un valor par" con probabilidad  $3/6$ ". En este caso, el evento "par" no es atómico, sino un subconjunto  $A \subseteq \Omega$ , donde  $A = \{2, 4, 6\}$ , es decir, los elementos en  $\Omega$  que satisfacen la proposición lógica "devuelven un valor par". Lo formalizamos de la siguiente manera:

$$p(A) = \frac{|A|}{|\Omega|} .$$

**Axiomas de probabilidad** . Dado un evento atómico  $\omega \in \Omega$ , su probabilidad  $p(\omega)$  es una función  $p : \Omega \rightarrow [0, 1]$  satisfactorio:

$$0 \leq p(\omega) \leq 1 \text{ and } \sum_{\omega \in \Omega} p(\omega) = 1 .$$

Además para **eventos no atómicos** Atenemos:

$$p(A) = \sum_{\omega \in A} p(\omega) = \sum_{\omega \in \Omega} p(\omega)[\omega \in A] ,$$

dónde  $p(\omega)[\omega \in A]$  lee:  $p(\omega)$  si  $\omega \in A$  y 0 de lo contrario.

Finalmente, para una secuencia contable de eventos **inconexos**  $A_1, A_2, \dots$  tenemos el siguiente axioma:

$$p(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} p(A_i) .$$

lo cual es consecuencia del PEI. En realidad, si los eventos no son disjuntos debemos considerar la definición completa del PEI (excluyendo-incluyendo intersecciones).

**Más propiedades** . Como resultado de los axiomas antes mencionados, tenemos:

$$p(\emptyset) = 0 \text{ and } p(\Omega) = 1 .$$

[Saltar al contenido principal](#)

$p(\bar{A}) = 1 - p(A)$  (complement) and  $A \subseteq B \Rightarrow p(A) \leq p(B)$  (monotonicity)

Finalmente, tenemos el PEI *binario*:

$$p(A \cup B) = p(A) + p(B) - p(A \cap B).$$

## 3.1. Ensayos independientes

### 3.1.1. Monedas y dados

Los eventos pueden verse como el resultado de un experimento determinado. Uno de los experimentos más sencillos es *lanzar una moneda*. Hay dos salidas posibles.  $\Omega = \{H, T\}$ . Si la moneda es *justa*, cada uno de estos resultados es *equiprobable* o *igualmente probable*:  $p(H) = p(T) = 1/2$ .

Ahora supongamos que el experimento de la moneda se puede repetir.  $n$  veces **en las mismas condiciones**. Cada una de estas repeticiones se llama **prueba**. Entonces, una pregunta natural a responder es "*¿cuál es la probabilidad de obtener, digamos?  $k$  cabezas tras  $n$  ensayos?*". Podemos usar combinatoria para responder a esta pregunta. En realidad, la solución es:

$$P(n, k) = \frac{\binom{n}{k}}{2^n}.$$

El número *total de casos* es  $2^n$ . si asignamos 1 a  $H$  y 0 a  $T$ , tenemos  $2^n$  números posibles de  $n$  bits (permutaciones con repetición) y cada uno de estos números concatena los resultados de  $n$  ensayos. Además, sólo  $\binom{n}{k}$  los casos son *favorables* ya que tenemos  $\binom{n}{k}$  grupos de  $k$  cabezas adentro de  $n$  ensayos.

En realidad, podemos entender mejor el razonamiento anterior reformulando  $P(n, k)$  como:

$$P(n, k) = \binom{n}{k} \left(\frac{1}{2}\right)^n.$$

Esto significa que la probabilidad de cualquier prueba es  $1/2$  y todo el  $n$  ensayos son **independientes**. Intuitivamente, independencia estadística significa que un ensayo no influye en el siguiente (las mismas condiciones experimentales para cualquier ensayo). Más formalmente,  $n$  eventos  $A_1, A_2, \dots, A_n$  son independientes si

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = p(A_1)p(A_2) \dots p(A_n) = \prod_{i=1}^n p(A_i).$$

Entonces, la probabilidad de obtener una secuencia dada dice  $HHHTT$  ( $n = 5$ ) es

$$p(HHHTT) = p(H)p(H)p(H)p(T)p(T) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2^5} = \frac{1}{32}.$$

En realidad, todas las secuencias con  $n = 5$  son equiprobables. Sin embargo, lo que hace  $HHHTT$  diferente de los demás es el hecho de que *tenemos 3 Hs* (y en consecuencia  $5 - 3 = 2$  cabezas). Si queremos resaltar este hecho, deberíamos *agrupar* todas las secuencias de  $n = 5$  ensayos teniendo 3 Hs (o equivalente 2 Ts). Algunas sugerencias:

- *El orden importa*. Estrictamente hablando, secuencias como  $HHHTTyTTHHH$  debe contar dos veces. Desde  $HyT$  se puede repetir (tres veces y dos veces respectivamente). tenemos **permutaciones con repetición**:

$$P_\sigma(n) = \frac{n!}{n_1!n_2!} \text{ where } n = 5, n_1 = k, n_2 = n - k$$

- Sin embargo, sabemos que

$$P_\sigma(n) = \frac{n!}{n_1!n_2!} = \frac{n!}{k!(n-k)!} = \binom{n}{k}$$

De aquí vienen las combinaciones: tenemos  $n = 5$  puestos y necesitamos cubrir 3 de ellos (el orden no importa) con cabezas. Entonces nosotros tenemos  $\binom{5}{3}$  formas de hacerlo. Funciona porque las cuerdas 11100, 00111 representan diferentes

[Saltar al contenido principal](#)

Por tanto, hemos utilizado la regla del producto para descomponer el problema en dos partes:

- Calcule la probabilidad de una configuración:  $p(HHHTT)$ .
- Calcule cuántas configuraciones diferentes tiene:  $\binom{5}{3}$ .

Entonces, la probabilidad  $p(\#H = 3, 5)$  de tener 3 cabezas adentro 5 ensayos es:

$$p(\#H = 3, 5) = \binom{5}{3} p(HHHTT) = \binom{5}{3} \frac{1}{32} = \frac{10}{32}.$$

En la práctica, utilice permutaciones con repeticiones en lugar de combinaciones cuando los resultados de un experimento sean más que 2. Ilustramos este caso en el siguiente ejercicio.

**Ejercicio**. Considere el problema de tirar un dado.  $n = 6$  corta en dados simultáneamente. ¿Cuál es la probabilidad de obtener resultados diferentes? ¿Y todos iguales?

Esto es equivalente a  $n$  pruebas de dados independientes. Desde  $|\Omega| = 6$ , tenemos  $\text{esop}(i) = 1/6$  para  $i = 1, 2, \dots, 6$ . Por lo tanto, la probabilidad de una secuencia dada de las pruebas son  $(1/6)^n$ . Este es el primer factor de la regla del producto (la probabilidad de una configuración determinada).

El segundo factor (el número de configuraciones posibles) surge de darse cuenta de que cada una de las  $n$  posiciones pueden ser ocupadas por diferentes valores. Como el orden importa, tenemos que contar el número de permutaciones *sin repetición o permutaciones con repetición individual* (los elementos deben ser diferentes) de  $n$  elementos. Esto lleva a  $n!$

$$p(\text{All\_diff.}) = n! \frac{1}{6^n} = \frac{n!}{1111111111} \cdot \frac{1}{6^n} = \frac{6!}{6^6} = \frac{6}{6} \cdot \frac{5}{6} \cdot \frac{4}{6} \cdot \frac{3}{6} \cdot \frac{2}{6} \cdot \frac{1}{6} = 0.015. \text{ Sin embargo, hay } n = 6 \text{ configuraciones donde todos los dados dan el mismo resultado: } p(\text{All\_equal.}) = \frac{n}{6^n} = \frac{1}{6^5} = 0.0001.$$

## 3.1.2. La distribución binomial

### 3.1.2.1. Unimodalidad

Como hemos visto en el apartado anterior, a la hora de realizar  $n$  ensayos independientes, las probabilidades de que ocurran algunos eventos son mayores que las de otros. En particular, **eventos extremos**  $E$  como maximizar (o minimizar) la apariencia de uno de los elementos de  $\Omega$  tiene la menor probabilidad. Sin embargo, dado que las probabilidades de todos los eventos se suman 1, la mayor parte de la *masa de probabilidad* debería residir en eventos no extremos.

Para ver eso, revisamos el lanzamiento de moneda, donde los dos eventos extremos son  $E = \{\text{All\_Ts}, \text{All\_Hs}\}$ , ambos con probabilidad  $1/2^n$ . Entonces nosotros tenemos

$$P(\text{All\_Ts}) = \binom{n}{0} \frac{1}{2^n} = \binom{n}{n} \frac{1}{2^n} = P(\text{All\_Hs}) = \frac{1}{2^n}.$$

Como resultado, la probabilidad de eventos no extremos se vuelve casi 1 cuando crece  $n$ :

$$P(\bar{E}) = 1 - P(E) = 1 - \frac{2}{2^n} = \frac{2^n - 2}{2^n} = \frac{2(2^{n-1} - 1)}{2^n} = \frac{2^{n-1} - 1}{2^{n-1}}.$$

Siendo todas las configuraciones particulares equiprobables (es decir  $1/2^n$ ), el hecho de que  $\lim_{n \rightarrow \infty} P(\bar{E}) = 1$  se debe al número de veces que se repite cada configuración particular.

Una mirada más cercana a  $2^n = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}$  nos da la respuesta. La probabilidad de tener  $k$  *Hs* se convierte en:

$$P(n, k) = \frac{\binom{n}{k}}{\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{n}},$$

y debido a la simetría de  $\binom{n}{k} = \binom{n}{n-k}$ , esta probabilidad va a crecer de  $k = 0$  hasta un determinado  $k = k_{\max}$  y luego disminuir para  $k = n$ . De hecho:

- Si incluso,  $k_{\max} = \frac{n}{2}$  y  $\binom{n}{0} < \binom{n}{1} < \dots < \binom{n}{\frac{n}{2}} > \binom{n}{\frac{n}{2}+1} > \dots > \binom{n}{n}$ .

[Saltar al contenido principal](#)

Es decir, por n impar tenemos un doble máximo de probabilidad. De todos modos la **distribución de probabilidad** entre los valores  $k = 0, 1, \dots, n$  es **unimodal**. Además, el aumento de la probabilidad de  $k - 1$  kantes de que el máximo esté dado por

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n-(k-1)}{k} \underbrace{\frac{n!}{(k-1)!(n-(k-1))(n-k)!}}_{[n-(k-1)]!} = \frac{n-(k-1)}{k} \binom{n}{k-1}$$

es decir

$$\frac{\binom{n}{k}}{\binom{n}{k-1}} = \frac{n-(k-1)}{k}.$$

Estas propiedades se pueden entender mejor estudiando el triángulo de Pascal.

### 3.1.2.2. Triángulo de Pascal

El [Triángulo de Pascal](#) ha tenido muchos nombres a lo largo de la historia de las matemáticas (por ejemplo, Triángulo de Tartaglia). Esta construcción da la  $\binom{n}{k}$  para todos  $n$ . Por ejemplo, en [HT](#), cada columna denota un valor y los coeficientes en esta columna son los  $n + 1$  los llamados **coeficientes binomiales** para tales  $n: \binom{n}{0}, \binom{n}{1}, \dots, \binom{n}{n}$ .

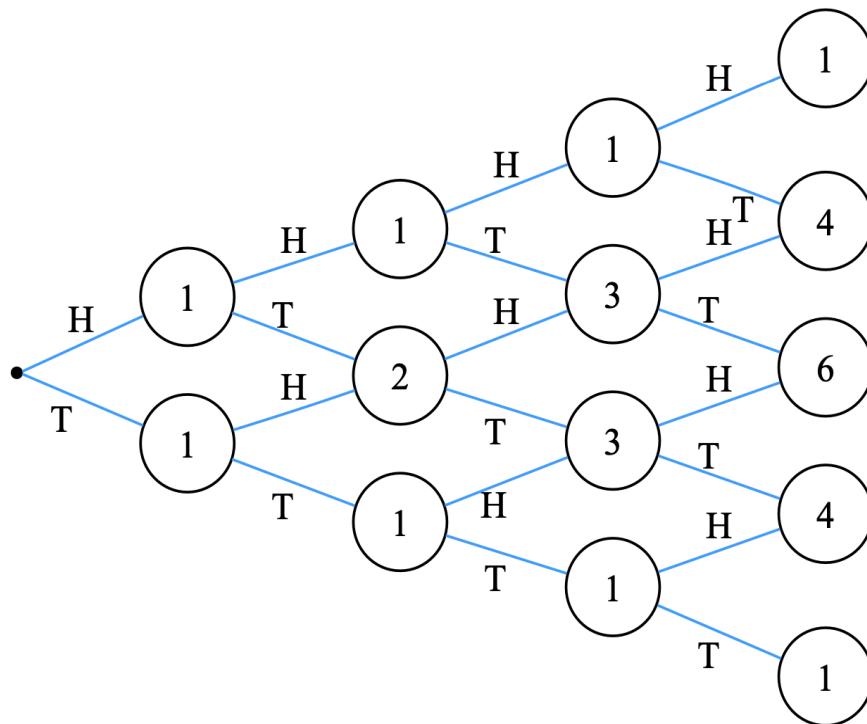


Fig. 3.1 Coeficientes binomiales.

El triángulo se construye de la siguiente manera:

- Comienza por  $n = 0$  y hacer una prueba. Tenemos  $\binom{1}{0} = 1$  formas de obtener un  $H$  y  $\binom{1}{1} = 1$  formas de conseguir un  $T$ . Luego establezcan  $n = 1$ .
- De  $H$  tenemos de nuevo 1 forma de obtener una  $H$  y una forma de obtener una  $T$ . Sin embargo, esto también sucede desde  $T$ . Por lo tanto, después del segundo experimento tenemos 4 posibles resultados:  $HH, HT, TH, TT$ . Dos de estos resultados son eventos extremos y dos de ellos colapsan en la misma representación.  $HT, TH$  (dos formas de conseguir uno  $H$  y uno  $T$ ). De esta manera el nodo central tiene un valor 2.
- Luego, fijamos  $n = 2$  y continúa...

Algunas propiedades:

- Como se señaló anteriormente, las columnas pares tienen coeficientes máximos únicos, mientras que las columnas impares tienen dos.

[Saltar al contenido principal](#)

- Si se normaliza el  $n$ -a columna por  $2^n$  tenemos la **distribución de probabilidad discreta** asociada a tener  $k = 0, 1, 2, \dots, n$  cabezas  $H$ s.
- **Los eventos extremos** siempre se ubican (por simetría) en las diagonales principales.
- **Eventos no extremos o regulares** comienzan a llenar las distribuciones a medida que  $n$  aumenta.
- Los coeficientes de cualquier **evento regular** se pueden obtener sumando los de sus padres en el árbol, ya que:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} .$$

- El coeficiente binomial en un nodo da el **número de caminos** que llegan a ese nodo desde el origen (equivalente a  $\# \Gamma$  en una cuadrícula sin obstáculos). Volveremos sobre este hecho más adelante.
- Comonumente, se vuelve cada vez más claro que la **distribución de probabilidad binomial** se centra en  $n/2$ , donde la probabilidad es máxima y luego disminuye a dos colas correspondientes a los eventos extremos. Esto nos permite entender intuitivamente el concepto de **valor medio**, tal y como lo definiremos más adelante.

Véase, por ejemplo **Bern**, donde resaltamos los coeficientes binomiales después  $n = 8$  ensayos. Compruebe que el coeficiente más alto en el nivel  $n = 8$  está en posición  $n/2$  (comenzando por 0).

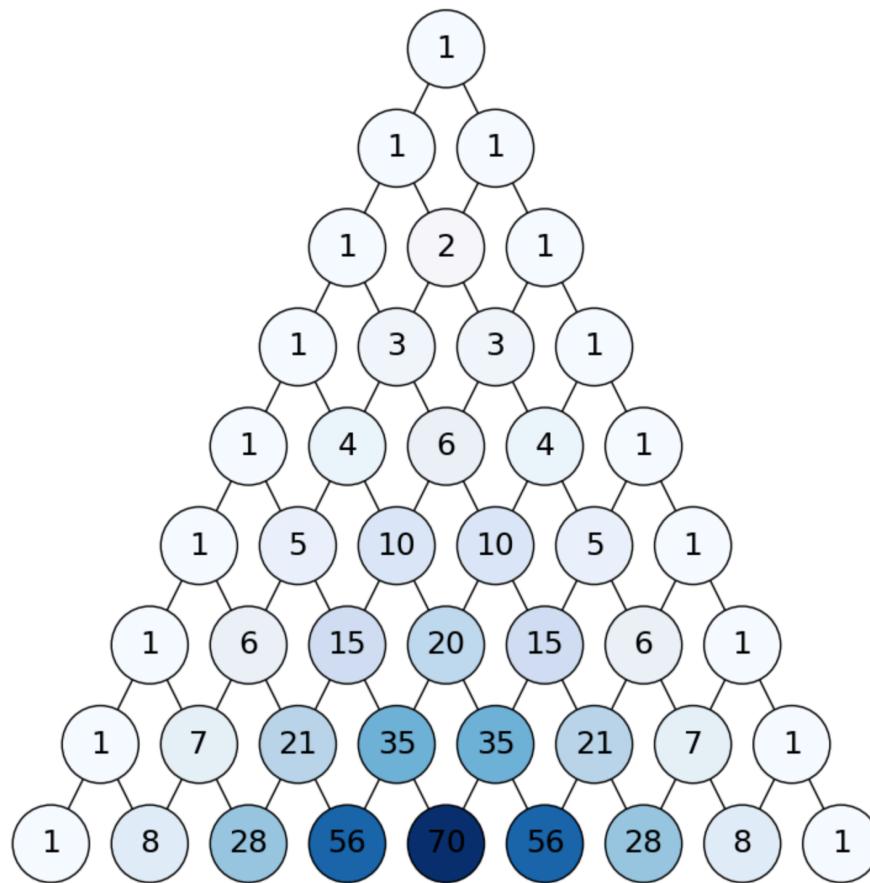


Fig. 3.2 Coeficientes binomiales.

¡Expresemos todas estas cosas en términos probabilísticos!

### 3.1.2.3. Valores probables y fluctuaciones

Uno de los elementos mágicos detrás del triángulo de Pascal es que proporciona los coeficientes binomiales, independientemente de si tenemos una **moneda justa** o no.

La moneda justa es un *caso particular* donde la **probabilidad de éxito** en un ensayo independiente (llamado **ensayo de Bernouilli**) es  $p = 1/2$  (Aquí entendemos por éxito "aterrizar de cabeza"). En consecuencia, la **probabilidad de falla** ("aterrizar en una cola") es  $q = 1 - p = 1/2$ . Entonces, la probabilidad de  $k$  éxitos después  $n$  ensayos generalmente están dados por

[Saltar al contenido principal](#)

$$P(n, k) = \binom{n}{k} p^k (1-p)^{n-k} = \binom{n}{k} p^k q^{n-k}.$$

Entonces,  $P(n, k)$  es la **probabilidad** de tener  $k$  éxitos en  $n$  ensayos. Esta es la línea sólida en  $\text{PD}$  donde hemos actuado 10,000 juegos, cada uno con  $n = 1,000$  y  $p = 1/2$ . En el eje que colocamos  $k = 0, 1, \dots, n$  (las hojas adentro  $\text{Bern}$  para  $n = 1,000$ ). En el eje trazamos (línea continua) el **teórico**  $P(n, k)$  valores. Pero también trazamos (idealmente) una barra por  $k$  valor. La altura de las barras es similar a  $P(n, k)$  pero no es idéntico. ¿Por qué? Porque hemos realizado 10,000 juegos. La altura de la barra  $k \times 10,000$  es el número de juegos donde hemos obtenido  $k$  salidas de  $n = 1,000$  ensayos. Este número (llamado **número observado** de éxitos) está cerca de  $10,000 \cdot P(n, k)$  pero **fluctúa** a su alrededor (a veces es más grande y otras más bajo).

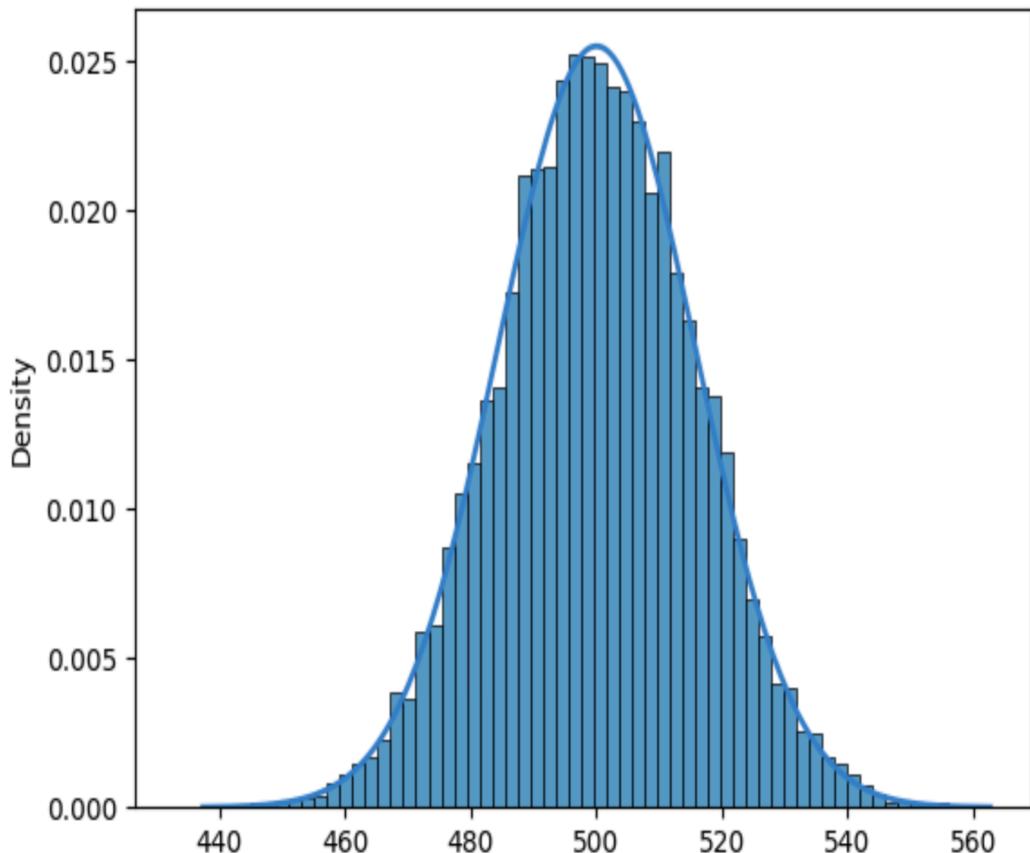


Fig. 3.3 Probabilidad teórica versus fluctuaciones.

Obviamente, el valor **más probable** o más probable de  $k$  es  $n/2 = 500$ . Sin embargo, no es tan obvio por qué los con menor valor distinto de cero  $P(n, k)$  están cerca de 440 y 560. Intuitivamente, esto está relacionado con los eventos extremos, pero estos valores merecen una interpretación matemática más profunda.

### 3.1.2.4. Expectativa y variación

**Variables aleatorias**. En el ejemplo anterior, cada uno de los 10,000 experimentos consisten en  $n = 1,000$  ensayos independientes, cada uno de los cuales resultó en aterrizar en  $H$ s o  $T$ s con probabilidad  $p$ . Entonces el **espacio muestral**  $\Omega$  es  $\{H, T\}^n$ . Bueno, una **variable aleatoria**.  $X$  es una función  $X : \Omega \rightarrow \mathbb{R}$ :  $X(\omega), \omega \in \Omega$ , como el **número de  $H$ s**. Esto está indicado por  $X(\omega) = k$ , y  $X \sim B(n, p)$  indica que  $X$  sigue una distribución de Bernoulli o Binomial.

**Expectativa**. La expectativa de una variable aleatoria  $X$  se define de la siguiente manera:

$$E(X) = \sum_{\omega \in \Omega} X(\omega) \cdot p(\omega) = \sum_x x \cdot p(X = x).$$

Para  $X \sim B(n, p)$ , tenemos:

[Saltar al contenido principal](#)

$$\begin{aligned}
E(X) &= \sum_{k=0}^n k \cdot P(n, k) \\
&= \sum_{k=0}^n k \cdot \binom{n}{k} p^k (1-p)^{n-k} \\
&= \sum_{k=0}^n n \binom{n-1}{k-1} p^k (1-p)^{n-k} \text{ since } k \binom{n}{k} = n \binom{n-1}{k-1} \\
&= n \sum_{k=0}^n \binom{n-1}{k-1} p^k (1-p)^{n-k} \\
&= np \sum_{k=1}^n \binom{n-1}{k-1} p^{k-1} (1-p)^{(n-1)-(k-1)} \\
&= np \sum_{r=0}^{n-1} \binom{n-1}{r} p^r (1-p)^{(n-1)-r} \text{ with } r = k-1 \\
&= np \text{ due to the Binomial Theorem .}
\end{aligned}$$

**El teorema del binomio** . El teorema del binomio de Newton está detrás del triángulo de Pascal y de la distribución binomial. Básicamente establece que los coeficientes de expansión  $(x + y)^n$ , donde  $n$  es un número entero, son los coeficientes binomiales  $\binom{n}{k}$ , como sigue:

$$\begin{aligned}
(x + y)^n &= \sum_{j=0}^n \binom{n}{j} x^{n-j} y^j \\
&= \binom{n}{0} x^n + \binom{n}{1} x^{n-1} y + \binom{n}{2} x^{n-2} y^2 + \dots + \binom{n}{n-1} x y^{n-1} + \binom{n}{n} y^n .
\end{aligned}$$

Observemos el teorema para  $n = 0, 1, 2, \dots$

- Para  $n = 0$ , tenemos  $(x + y)^0 = \binom{0}{0} x^{0-0} y^0 = 1$ .
- Para  $n = 1$ , tenemos  $(x + y) = \binom{1}{0} x + \binom{1}{1} y = 1 \cdot x + 1 \cdot y$ .
- Para  $n = 2$ ,  $(x + y)^2 = \binom{2}{0} x^2 + \binom{2}{1} x y + \binom{2}{2} y^2 = 1 \cdot x^2 + 2 \cdot x y + 1 \cdot y^2$ .
- Para  $n = 3$ ,  $(x + y)^3 = \binom{3}{0} x^3 + \binom{3}{1} x^2 y + \binom{3}{2} x y^2 + \binom{3}{3} y^3 = 1 \cdot x^3 + 3 \cdot x^2 y + 3 \cdot x y^2 + 1 \cdot x^3$ .

Si observas los coeficientes, están dados por los niveles  $n = 0, 1, 2, 3, \dots$  del Triángulo de Pascal! En otras palabras, aquí los **eventos extremos** son los coeficientes unitarios de  $x^n y y^n$  respectivamente. Luego, la correspondiente  $\binom{n}{k}$  coeficiente de  $x^{n-k} y^k$  solo indica el producto correspondiente, es decir cuantos  $x$ s y cuantos  $y$ s. ¿Cómo tomamos cada término?

Como resultado, la forma en que se construye el triángulo de Pascal juega un papel fundamental en la **demonstración inductiva** del teorema. Simplemente recuerda la expresión.  $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$  ..

De todos modos, volviendo a la expectativa de  $E(X) = np$ , simplemente aplica el teorema que expresa  $p^r (1-p)^{(n-1)-r}$  como  $p^r q^{(n-1)-r}$  para el coeficiente binomial  $\binom{n-1}{r}$ . Todo lo que estamos haciendo es computar  $(x + y)^{n-1}$  dónde  $x + y = p + q = 1$ .

Como resultado:

$$\sum_{r=0}^{n-1} \binom{n-1}{r} p^r q^{(n-1)-r} = (p + q)^{n-1} = 1^{n-1} = 1.$$

**Variación** . El hecho de que cuando  $X \sim B(n, p)$  tenemos  $E(X) = np$  nos da directamente el número más probable de éxitos (ver **PD**). Sin embargo, para explicar la **cantidad de desviación de la expectativa** nos basamos en la varianza que se define de la siguiente manera:

$$Var(X) = \sum_x (x - E(X))^2 \cdot p(X = x) .$$

De este modo,  $Var(X)$  es la expectativa de las desviaciones cuadradas de  $E(X)$ :

[Saltar al contenido principal](#)

$$\begin{aligned}
 Var(X) &= E([X - E(X)]^2) \\
 &= E(X^2 + E(X)^2 - 2XE(X)) \\
 &= E(X^2) + E(E(X)^2) - 2E(XE(X)) \\
 &= E(X^2) + E(X)^2 - 2E(X)E(X) \\
 &= E(X^2) + E(X)^2 - 2E(X)^2 \\
 &= E(X^2) - E(X)^2 .
 \end{aligned}$$

Una forma interesante (y sencilla) de calcular  $Var(X)$  para variables binomiales  $X$  es darse cuenta de que  $X \sim B(n, p)$  significa que  $X = \sum_{i=1}^n Y_i$  donde  $Y_i \sim Bern(p)$ , es decir, **variables de Bernouilli** donde los resultados pueden ser 1 (éxito) o 0 (falla).

En particular, si  $Y \sim Bern(p)$  tenemos:

$$E(Y) = \sum_y y \cdot p(Y = y) = 1 \cdot p(Y = 1) + 0 \cdot p(Y = 0) = p .$$

Desde  $X$  es la suma de  $n$   $Y_i$ s:  $E(X) = E(Y_1) + E(Y_2) + \dots + E(Y_n)$ . Como [la expectativa de una suma es la suma de las expectativas \(independientemente de si las variables son independientes o no\)](#), tenemos:

$$E(X) = E\left(\sum_{i=1}^n Y_i\right) = \sum_{i=1}^n E(Y_i) = np .$$

Ahora, para la informática.  $E(Y^2), Y \sim Bern(p)$ , hacemos:

$$E(Y^2) = \sum_y y^2 \cdot p(Y = y) = 1^2 \cdot P(Y = 1) + 0^2 \cdot P(Y = 0) = p .$$

Como resultado

$$Var(Y) = E(Y^2) - E(Y)^2 = p - p^2 = p(1 - p) = pq .$$

Ahora aproveche el hecho de que [la varianza de la suma es la suma de las varianzas cuando las variables son independientes](#). Como este es el caso de  $Y_i \sim Bern(p)$ . Entonces podemos calcular

$$Var(X) = Var(Y_1 + Y_2 + \dots + Y_n) = \sum_{i=1}^n Var(Y_i) = npq ;$$

Eso es obvio  $Var(X)$  aumenta linealmente con  $n$ . Esto simplemente significa que la forma de la distribución (ver [PD](#)) se vuelve cada vez más ancha a medida que aumenta? Realmente no, es la varianza lo que aumenta. Más adelante veremos que por  $n \rightarrow \infty$  tenemos una distribución gaussiana.

**Ejercicio** . Es importante darse cuenta de que las variables binomiales son en realidad sumas de **ensayos de Bernouilli** . Pero, ¿qué se puede considerar realmente un proceso de Bernouilli? En principio, un ensayo de Bernouilli se refiere a cualquier experimento cuyo resultado pueda agruparse en dos **grupos mutuamente excluyentes** : "rojo versus negro", "rojo versus (negro o verde)", "par versus impar", " $x \leq k$  vs  $x > k$ ", etc.

Consideremos, por ejemplo, el siguiente ensayo de Bernouilli: **Lanzar 2 corta en dados simultáneamente** y registra los resultados como  $(x, y)$ . Entonces infiere  $p + q = 1$  en los siguientes casos:

- a)  $p = \{\text{Prob. of } x + y \text{ is even}\}$
- b)  $p = \{\text{Prob. of } x \times y \text{ is even}\}$
- c)  $p = \{\text{Prob. of } x + y > 7 \text{ is even}\}$

y proporcione también las **expectativas y varianzas** de las variables de Bernouilli asociadas con estos experimentos.

En primer lugar, el espacio muestral.  $\Omega$  viene dado por el producto cartesiano  $\Omega = \{1, \dots, 6\} \times \{1, \dots, 6\}$ . Así, tenemos

[Saltar al contenido principal](#)

$$\Omega = \begin{matrix} (1,1) & (1,2) & (1,3) & (1,4) & (1,5) & (1,6) \\ (2,1) & (2,2) & (2,3) & (2,4) & (2,5) & (2,6) \\ (3,1) & (3,2) & (3,3) & (3,4) & (3,5) & (3,6) \\ (4,1) & (4,2) & (4,3) & (4,4) & (4,5) & (4,6) \\ (5,1) & (5,2) & (5,3) & (5,4) & (5,5) & (5,6) \\ (6,1) & (6,2) & (6,3) & (6,4) & (6,5) & (6,6) \end{matrix}$$

$$\Omega_+ = \begin{matrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 7 & 8 & 9 & 10 & 11 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{matrix}$$

$$\Omega_\times = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 6 & 8 & 10 & 12 \\ 3 & 6 & 9 & 12 & 15 & 18 \\ 4 & 8 & 12 & 16 & 20 & 24 \\ 5 & 10 & 15 & 20 & 25 & 30 \\ 6 & 12 & 18 & 24 & 30 & 36 \end{matrix}$$

dónde  $\Omega_+$  y  $\Omega_\times$  son los espacios muestrales de las variables aleatorias  $X = \{x + y : (x, y)\}$  y  $Y = \{x \times y : (x, y)\}$ . **a)** Para cada fila de

$\Omega_+$  tenemos 3 incluso y 3 impares. Como resultado,  $p = q = 1/2$ . Entonces

$$E(X_{even}) = 1 \cdot p + 0 \cdot q = p = \frac{1}{2}$$

$$Var(X_{even}) = E(X_{even}^2) - E(X_{even})^2 = p - p^2 = p(1 - p) = pq = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

**b)** Las filas impares de  $\Omega_\times$  tienen 3 incluso y 3 extraño. Sin embargo, las filas pares tienen 6 (todos) pares y 0 (ninguno) extraño. Como resultado, tenemos  $3 \times 3 + 3 \times 0 = 9$  resultados extraños y  $3 \times 3 + 3 \times 6 = 27$  de 36 resultados. Entonces nosotros tenemos  $p = 27/36 = 3/4$  y  $q = 9/36 = 1/4$ . Entonces

$$E(Y_{even}) = 1 \cdot p + 0 \cdot q = p = \frac{3}{4}$$

$$Var(Y_{even}) = E(Y_{even}^2) - E(Y_{even})^2 = p - p^2 = p(1 - p) = pq = \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{16} = 0.1875$$

**c)** Para esta parte necesitamos calcular el dominio de  $X = \{x + y : (x, y)\}$ , cual es  $X = \{2, 3, \dots, 12\}$ . También necesitamos saber las **frecuencias** de cada valor. Las **frecuencias** de las sumas vienen dadas por las simetrías de este operador:

$$\begin{aligned} X(2) &= X(1, 1) = 1 \\ X(3) &= X(1, 2) = X(2, 1) = 2 \\ X(4) &= X(1, 3) = X(3, 1) = X(2, 2) = 3 \\ X(5) &= X(1, 4) = X(4, 1) = X(2, 3) = X(3, 2) = 4 \\ X(6) &= X(1, 5) = X(5, 1) = X(2, 4) = X(4, 2) = X(3, 3) = 5 \\ X(7) &= X(1, 6) = X(6, 1) = X(2, 5) = X(5, 2) = X(3, 4) = X(4, 3) = 6 \\ X(8) &= X(2, 6) = X(6, 2) = X(3, 5) = X(5, 3) = X(4, 4) = 5 \\ X(9) &= X(5, 4) = X(4, 5) = X(6, 3) = X(3, 6) = 4 \\ X(10) &= X(6, 4) = X(4, 6) = X(5, 5) = 3 \\ X(11) &= X(6, 5) = X(5, 6) = 2 \\ X(12) &= X(6, 6) = 1 \end{aligned}$$

Entonces,  $Z_{x+y>7}$  tiene  $5 + 4 + 3 + 2 + 1 = n(n + 1)/2 = 5 \cdot (5 + 1)/2 = 15$  resultados favorables de 36 posibilidades. Como resultado, tenemos  $p = 15/36 = 5/12$  y  $q = 21/36 = 7/12$ . Entonces

[Saltar al contenido principal](#)

$$E(Z_{x+y>7}) = 1 \cdot p + 0 \cdot q = p = \frac{5}{12} = 0.42$$

$$\text{Var}(Z_{x+y>7}) = E(Z_{x+y>7}^2) - E(Z_{x+y>7})^2 = p - p^2 = p(1-p) = pq = \frac{15}{36} \cdot \frac{21}{36} = 0.24$$

### 3.1.2.5. Desigualdades fundamentales

Los eventos extremos tienen pequeñas probabilidades. En  $X \sim B(n, p)$  la probabilidad decae desde su valor máximo (por  $k = \lfloor np \rfloor$ ) hasta cerca de cero en los eventos extremos. Tal decadencia es algo descrita por  $\text{Var}(X)$ . Sin embargo, tenemos que profundizar más para caracterizar la **probabilidad de eventos raros**.

En primer lugar, deberíamos medir cómo decae la probabilidad a medida que nos alejamos de la expectativa. Para esta tarea, nos basamos nuevamente en mirar  $X \sim B(n, p)$  como la suma de juicios de Bernoulli  $Y_i \sim \text{Bern}(p)$ .

Para  $X = Y_1 + Y_2 + \dots + Y_n$ , tenemos

$$p\left(\sum_{i=1}^n Y_i - E(X)\right) = p(X - E(X))$$

Límites del **teorema de Hoeffding**:  $p(X - E(X) \geq t)$  para  $t > 0$  y  $X$  siendo la suma de variables independientes  $Y_i$  satisfactorio  $a_i \leq Y_i \leq b_i$  casi con seguridad o como (es decir, con probabilidad 1), como sigue:

$$\begin{aligned} p(X - E(X) \geq t) &\leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right) \\ p(|X - E(X)| \geq t) &\leq 2 \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right). \end{aligned}$$

Este teorema simplemente significa que desviar set unidades de  $E(X)$ , da como resultado una **caída exponencial**. Esto justifica la forma exponencial de la distribución binomial a medida que pasamos de  $E(X)$ ; hacia los eventos extremos! En particular, para esta distribución, donde  $0 \leq Y_i \leq 1$  como, los límites de Hoeffding son:

$$p(X - np \geq t) \leq \exp\left(-\frac{2t^2}{n}\right) \quad \text{and} \quad p(|X - np| \geq t) \leq 2 \exp\left(-\frac{2t^2}{n}\right).$$

Curiosamente, la caída exponencial se ve atenuada por  $n$ : el mas lento es el decaimiento ya que la distribución se vuelve cada vez más plana a medida que aumenta.

**Distribución acumulativa**. Hasta ahora nos hemos centrado en describir variables aleatorias en términos de caracterizar  $p(X = x)$  (*función punto-masa* o pmf). Sin embargo, a veces es útil cuantificar la **mayor parte de la probabilidad**, por ejemplo entre dos valores extremos  $a < b$ . Para  $X \sim B(n, p)$ , sabemos que este bulto es casi 1 (pero no *casi seguro* a menos que  $n \rightarrow \infty$ , ya que existen eventos extremos). Sin embargo, para  $k : a \leq k \leq b$ , ¿cuál es la probabilidad de que el número de caras sea "menor o igual que  $k$ "?

La forma habitual de responder a esta pregunta es calcular  $p(X \leq k)$ :

$$p(X \leq k) = \sum_{x \leq k} p(X = x).$$

Podemos utilizar la **cota de Hoeffding** para dar una idea de esta probabilidad, ya que

$$\begin{aligned} p(X \leq k) &= 1 - p(X > k) \\ &= 1 - p(X \geq k+1) \\ &= 1 - p(X - np \geq (k+1) - np) \\ &\geq \exp\left(-\frac{((k+1) - np)^2}{n}\right). \end{aligned}$$

De hecho,  $p(X \leq k)$  es mucho mayor que el neg-exp. Curiosamente, cuando  $k \approx np$  (cerca del valor esperado),  $p(X \leq k) \geq \frac{1}{n}$ ,

[Saltar al contenido principal](#)

**El límite de Chernoff** es uno de los límites más estrictos para cuantificar la probabilidad de eventos raros. Está formulado de la siguiente manera. si tenemos  $n$  variables independientes  $Y_1, Y_2, \dots, Y_n$  con  $p(Y_i = 1) = p_i$  y  $p(Y_i = 0) = 1 - p_i$ ,  $X = \sum_{i=1}^n Y_i$  tiene expectativa  $E(X) = \sum_{i=1}^n p_i$  y tenemos:

$$\begin{aligned} \text{(Lower tail)} \quad p(X - E(X) \leq -\lambda) &\leq \exp\left(-\frac{\lambda^2}{2E(X)}\right) \\ \text{(Upper tail)} \quad p(X - E(X) \geq \lambda) &\leq \exp\left(-\frac{\lambda^2}{2(E(X) + \lambda/3)}\right). \end{aligned}$$

La formulación anterior del límite de Chernoff fue propuesta en el [documento de Survey que trata sobre las desigualdades de concentración](#). En realidad tenemos:

- El límite inferior de la cola es válido para todos  $\lambda \in [0, E(X)]$  y por lo tanto para todo real  $\lambda \geq 0$ .
- La parte superior de la cola también es válida para todos los reales  $\lambda \geq 0$ .

Tenga en cuenta que esta versión del límite de Chernoff decrece más lentamente que el límite de Hoeffding ya que el denominador está dominado por  $E(X)$ .

Calculemos ahora las probabilidades de las colas en [PD](#): 440 y 560. Desde  $E(X) = np = 500$ , tenemos eso  $440 - 500 = -60$  y  $560 - 500 = 60$ , establecemos  $\lambda = 60$ . Entonces

- $p(X - E(X) \leq -\lambda) \leq \exp\left(-\frac{60^2}{1000}\right) = 0.0027$ .
- $p(X - E(X) \geq \lambda) \leq \exp\left(-\frac{60^2}{2(500+60/3)}\right) = 0.0031$ .

Hasta ahora hemos caracterizado la distribución Binomial (concepto, expectativa, varianza y eventos raros). La distribución Binomial juega un papel fundamental en el análisis de algoritmos de IA para descubrir la mejor solución (siempre que sea posible). Estos algoritmos exploran un árbol de forma "inteligente". Antes de dar una intuición de este punto, introduzcamos un concepto importante también relacionado con la probabilidad y la "exploración": el **paseo aleatorio**. En particular, nos centraremos en los paseos aleatorios bajo la distribución binomial.

### 3.1.2.6. Paseos aleatorios

Comencemos definiendo el siguiente juego:

- Poner un viajero en  $x = 0$ .
- Tirar una moneda.
  - Si el resultado es movimiento de cabeza hacia la derecha:  $x = x + 1$ .
  - De lo contrario, muévase hacia la izquierda:  $x = x - 1$ .
- Continuar hasta llegar a  $n$  lanzamientos.

El procedimiento anterior describe un **paseo aleatorio unidimensional** a través del  $\mathbb{Z}$ . Uno de los propósitos de este juego es responder a la pregunta: "¿Hasta dónde llega en promedio el viajero?"

Un poco formalmente, el problema consiste en encontrar la expectativa de  $Z$ , la suma de  $n$  eventos independientes  $Y_i$  con salida tampoco +1 o -1. Por una moneda justa,  $E(Y_i) = 1 \cdot p + (-1) \cdot p = 0$  y esto implica que

$$E(Z) = E(\sum_{i=1}^n Y_i) = \sum_{i=1}^n E(Y_i) = 0.$$

En realidad, para ser justos, también podemos responder a la pregunta: "¿Cuál es la probabilidad de llegar a un número entero determinado?" [zdespuésn¿pasos?](#) . Esto se puede hacer simplemente consultando el Triángulo de Pascal! En otras palabras, este problema equivale a colocar el  $\mathbb{Z}$  línea en la parte superior del triángulo de Pascal y alineando  $x = 0$  con el vértice superior del triángulo. Eso lo hacemos en [PascalZ](#), para aclarar que:

[Saltar al contenido principal](#)

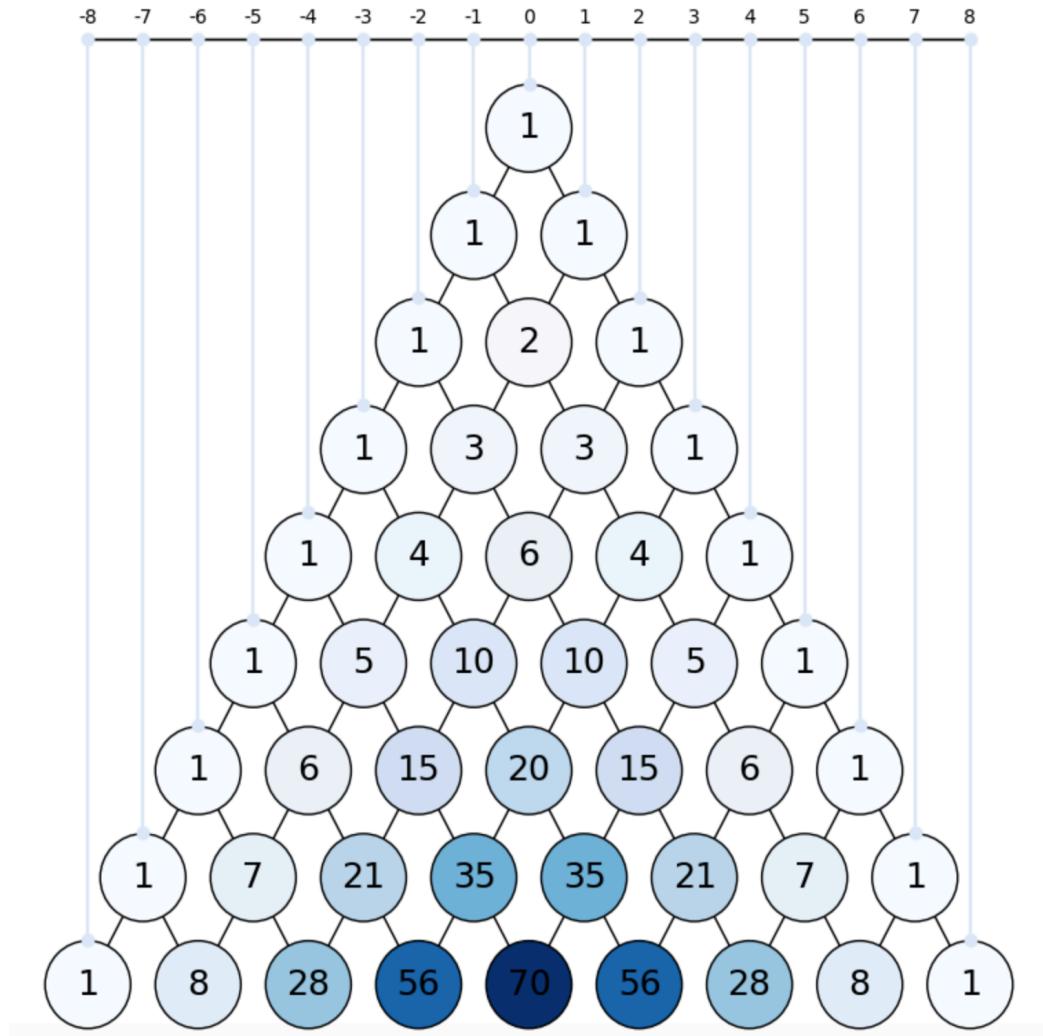


Fig. 3.4 Paseo aleatorio unidimensional (justo) sobre el triángulo de Pascal.

- El caminante aleatorio (RW) podrá avanzar hacia la izquierda (negativo), hacia la derecha (positivo) o regresar a casa (cero). Aterrizando en un número entero dado z solo la probabilidad  $P(z, n)$  de  $z \geq 0$  (éxitos) o  $z < 0$  (fallas).
- En [PascalZ](#), enlazamos  $z \in \mathbb{Z}$  con nodos de los respectivos *primeros niveles* donde tenemos  $z$  éxitos (fracasos), pero esta línea se extiende a los nodos debajo de ellos si se cumple esta propiedad: "si no tenemos  $z$  éxitos (fracasos) todavía, tal vez los tengamos más adelante". ¡Este es básicamente el conflicto del jugador!
- Sin embargo, a largo plazo se espera que el jugador aterrice en  $z = 0$  (ni ganar ni perder) si su riqueza es lo suficientemente grande. Esto es porque  $E(Z) = 0$  es equivalente a  $E(X) = np$  para  $X \sim B(n, p)$ .
- Por supuesto, la esperanza (aterrizar en  $z > 0$ ) o riesgo (aterrizar en  $z < 0$ ) se mide por la probabilidad de eventos extremos como explicamos anteriormente.
- Lógicamente, si los eventos extremos ocurren con mucha más frecuencia de lo que predice la teoría, uno puede pensar que la moneda está cargada (no es justo).

**Independientemente de la dirección**. hemos visto eso  $E(Z) = 0$  (Avanzar y retroceder es igualmente probable). Sin embargo, ¿qué sucede si reformulamos la pregunta original de la siguiente manera: "¿Hasta dónde llega en promedio el viajero, **independientemente de la dirección** ?". Responder a esta pregunta implica calcular

$$E(Z^2) = E\left[\left(\sum_{i=1}^n Y_i^2\right)\right].$$

Usando la identidad (para cualquier  $Z$  dado por la suma de variables **independientes distribuidas idénticamente** o iid):

$$\text{Var}(Z) = E(Z^2) - E(Z)^2.$$

[Saltar al contenido principal](#)

$$\begin{aligned}Var(Y) &= E(Y^2) - E(Y)^2 \\E(Y^2) &= (+1)^2 \cdot p + (-1)^2 \cdot q = p + q = 1 \\E(Y)^2 &= [(+1) \cdot p + (-1) \cdot q]^2 = [p - q]^2.\end{aligned}$$

Desde  $p = q$  por una moneda justa, tenemos  $Var(Y) = 1$ , un resultado  $Var(Z) = E(Z^2) = n$ . Por ejemplo, en **RWrand** trazamos 100 RW justos para  $n = 10,000$ . La oscuridad del azul de cada paseo es proporcional a  $Var(Z) = E(Z^2)$  (Hemos invertido este brillo con respecto a las cifras anteriores para visualizar mejor los caminos extremos). Tenga en cuenta que la mayoría de los caminos tienen "desviaciones" delimitadas superiormente por  $\pm 3\sqrt{Var(Z)} = \pm 3\sqrt{n} = \pm 300$ .

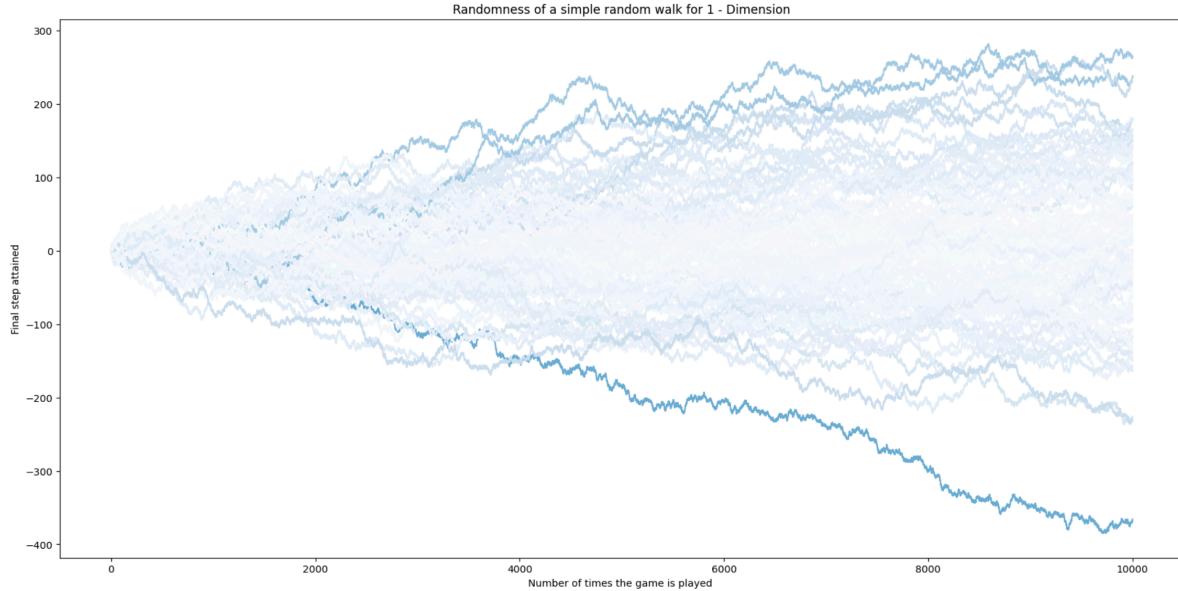


Fig. 3.5 Paseo aleatorio unidimensional (justo): desviaciones.

### 3.1.2.7. La distribución normal

Cuando  $n \rightarrow \infty$ , la naturaleza combinatoria de  $B(n, p)$  y sus vínculos con los paseos aleatorios se pueden describir de una manera más sencilla. El  $B(n, p)$  en el límite llega otra distribución: la conocida distribución *Normal* o *Gaussiana*.

En este sentido, explotamos la aproximación de Stirling reescrita como

$$n! \approx \sqrt{2\pi n} \cdot n^n e^{-n}.$$

Sustituimos esta fórmula en la probabilidad de  $k$  éxitos, para aproximar *todos los factoriales*:

$$\begin{aligned}p(X = k) &= \binom{n}{k} p^k q^{n-k} \\&= \frac{n!}{k!(n-k)!} p^k q^{n-k} \\&\approx \frac{\sqrt{2\pi n} \cdot n^n e^{-n}}{\sqrt{2\pi k} \cdot k^k e^{-k} \sqrt{2\pi(n-k)} \cdot (n-k)^{(n-k)} e^{-(n-k)}} p^k q^{n-k} \\&\approx \frac{\sqrt{2\pi n} \cdot n^n e^{-n}}{\sqrt{2\pi k} \sqrt{2\pi(n-k)} \cdot k^k \cdot (n-k)^{(n-k)} e^{-k} e^{-(n-k)}} p^k q^{n-k} \\&= \frac{\sqrt{2\pi n} \cdot n^n e^{-n}}{\sqrt{2\pi k} \sqrt{2\pi(n-k)} \cdot k^k \cdot (n-k)^{(n-k)} e^{-n}} p^k q^{n-k} \\&= \sqrt{\frac{n}{2\pi k(n-k)}} \cdot \frac{n^n}{k^k \cdot (n-k)^{n-k}} p^k q^{n-k} \\&= \sqrt{\frac{n}{2\pi k(n-k)}} \cdot \frac{n^k n^{n-k}}{k^k \cdot (n-k)^{n-k}} p^k q^{n-k} \\&= \sqrt{\frac{n}{2\pi k(n-k)}} \cdot \left(\frac{np}{k}\right)^k \cdot \left(\frac{nq}{n-k}\right)^{n-k}.\end{aligned}$$

[Saltar al contenido principal](#)

En este punto conviene formular el número de éxitos  $k$  en términos de desviaciones  $k = np + z$  de la expectativa  $np$  como lo hicimos al definir paseos aleatorios. Como resultado,

$$\begin{aligned} n - k &= n - (np + z) = n - (n(1 - q) + z) \\ &= n - (n - nq + z) \\ &= nq - z. \end{aligned}$$

y tenemos

$$\begin{aligned} p(X = k) &= p(X = np + z) \\ &\approx \sqrt{\frac{n}{2\pi(n - k)}} \cdot \left(\frac{np}{np + z}\right)^{np+z} \cdot \left(\frac{nq}{nq - z}\right)^{nq-z} \\ &= \sqrt{\frac{n}{2\pi(np + z)(nq - z)}} \cdot \left(\frac{np}{np + z}\right)^{np+z} \cdot \left(\frac{nq}{nq - z}\right)^{nq-z} \end{aligned}$$

Desde

$$\begin{aligned} \frac{n}{(np + z)(nq - z)} &= \frac{n}{n^2pq - npz + nqz - z^2} \\ &= \frac{1}{npq + \frac{-pz + qz - z^2}{n}} \\ &\approx \frac{1}{npq} \end{aligned}$$

tenemos

$$\begin{aligned} p(X = k) &= p(X = np + z) \\ &\approx \frac{1}{\sqrt{2\pi \cdot npq}} \cdot \left(\frac{np}{np + z}\right)^{np+z} \cdot \left(\frac{nq}{nq - z}\right)^{nq-z} \\ &= \frac{1}{\sqrt{2\pi \cdot npq}} \cdot \left(\frac{np + z}{np}\right)^{-(np+z)} \cdot \left(\frac{nq - z}{nq}\right)^{-(nq-z)} \\ &= \frac{1}{\sqrt{2\pi \cdot npq}} \cdot \left(\frac{np}{np + z}\right)^{np+z} \cdot \left(\frac{nq}{nq - z}\right)^{nq-z} \\ &= \frac{1}{\sqrt{2\pi \cdot npq}} \cdot \left(1 + \frac{z}{np}\right)^{-(np+z)} \cdot \left(1 - \frac{z}{nq}\right)^{-(nq-z)} \\ &= \frac{1}{\sqrt{2\pi \cdot npq}} \cdot \left(1 + \frac{z}{np}\right)^{-(np+z)} \cdot \left(1 - \frac{z}{nq}\right)^{-(nq-z)}. \end{aligned}$$

Ahora, para resaltar la forma exponencial de  $p(X = np + z)$  tomemos troncos a ambos lados

$$\begin{aligned} \log p(X = k) &= \log p(X = np + z) \\ &\approx \log \frac{1}{\sqrt{2\pi \cdot npq}} + \left(1 + \frac{z}{np}\right)^{-(np+z)} + \left(1 - \frac{z}{nq}\right)^{-(nq-z)} \\ &= C + \log \left(1 + \frac{z}{np}\right)^{-(np+z)} + \log \left(1 - \frac{z}{nq}\right)^{-(nq-z)} \\ &\approx C - (np + z) \log \left(1 + \frac{z}{np}\right) - (nq - z) \log \left(1 - \frac{z}{nq}\right) \\ &= C - A - B \end{aligned}$$

Ahora, explotamos las expansiones de Taylor de  $\log(1 + x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 \dots$  y  $\log(1 - x) = -x - \frac{1}{2}x^2 - \frac{1}{3}x^3 \dots$

Llevando a los términos cuadráticos, para A, B tenemos

[Saltar al contenido principal](#)

$$A = (np + z) \left[ \frac{z}{np} - \frac{1}{2} \left( \frac{z}{np} \right)^2 \right] \text{ and } B = (nq - z) \left[ -\frac{z}{nq} - \frac{1}{2} \left( \frac{z}{nq} \right)^2 \right]$$

Entonces nosotros tenemos

$$\begin{aligned} A &= \left[ np \frac{z}{np} - np \frac{1}{2} \left( \frac{z}{np} \right)^2 \right] + \left[ z \frac{z}{np} - z \frac{1}{2} \left( \frac{z}{np} \right)^2 \right] \\ &= z - \frac{1}{2} \frac{z^2}{np} + \frac{z^2}{np} - \frac{1}{2} \frac{z^3}{(np)^2} \\ &= z + \frac{1}{2} \frac{z^2}{np} - \frac{1}{2} \frac{z^3}{(np)^2} \end{aligned}$$

y de manera similar

$$\begin{aligned} B &= \left[ -nq \frac{z}{nq} - nq \frac{1}{2} \left( \frac{z}{nq} \right)^2 \right] - \left[ -z \frac{z}{nq} - z \frac{1}{2} \left( \frac{z}{nq} \right)^2 \right] \\ &= -z - \frac{1}{2} \frac{z^2}{nq} + \frac{z^2}{nq} + \frac{1}{2} \frac{z^3}{nq} \\ &= -z + \frac{1}{2} \frac{z^2}{nq} + \frac{1}{2} \frac{z^3}{(nq)^2} \end{aligned}$$

enchufar  $A, B, C$  en

$$\begin{aligned} \log p(X = k) &= \log p(X = np + z) \\ &\approx C - A - B \\ &= \log \frac{1}{\sqrt{2\pi \cdot npq}} - z - \frac{1}{2} \frac{z^2}{np} + \frac{1}{2} \frac{z^3}{(np)^2} + z - \frac{1}{2} \frac{z^2}{nq} - \frac{1}{2} \frac{z^3}{(nq)^2} \\ &= \log \frac{1}{\sqrt{2\pi \cdot npq}} - z - \frac{1}{2} \frac{z^2}{np} + \frac{1}{2} \frac{z^3}{(np)^2} - z - \frac{1}{2} \frac{z^2}{nq} - \frac{1}{2} \frac{z^3}{(nq)^2} \\ &= \log \frac{1}{\sqrt{2\pi \cdot npq}} - \frac{1}{2} \frac{z^2}{np} - \frac{1}{2} \frac{z^2}{nq} + \left( \frac{1}{2} \frac{z^3}{(np)^2} - \frac{1}{2} \frac{z^3}{(nq)^2} \right) \\ &\approx \log \frac{1}{\sqrt{2\pi \cdot npq}} - \frac{1}{2} \frac{z^2}{np} - \frac{1}{2} \frac{z^2}{nq} \\ &= \log \frac{1}{\sqrt{2\pi \cdot npq}} + \frac{(-q - p)z^2}{2npq} \\ &= \log \frac{1}{\sqrt{2\pi \cdot npq}} + \frac{(-q - (1 - q))z^2}{2npq} \\ &= \log \frac{1}{\sqrt{2\pi \cdot npq}} - \frac{z^2}{2npq} \end{aligned}$$

Como resultado, si reemplazamos  $z$  por  $k - np$  tenemos

$$\log p(X = k) - \log \frac{1}{\sqrt{2\pi \cdot npq}} = -\frac{z^2}{2npq},$$

es decir, el llamado **teorema de De Moivre-Laplace** produce la expresión habitual para la distribución Normal como límite de la Distribución Binomial :

$$p(X = k) = \frac{1}{\sqrt{2\pi \cdot npq}} \exp \left( -\frac{1}{2} \frac{(k - np)^2}{npq} \right),$$

De manera más general, compones la "media" del Binomio, se renombra como **media**  $\mu$  de la Normalidad. De manera similar, compones la "varianza"  $\sigma^2$  de la Normal, cuya raíz cuadrada es la **desviación estándar**

[Saltar al contenido principal](#)

Por lo tanto, podemos decir que  $X \sim N(\mu, \sigma^2)$  si su pmf es

$$p(X = x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right).$$

**La normalidad estandarizada**. Volviendo al hecho de que

$$k = np + z,$$

lo traducimos a

$$x = \mu + z, \text{ i.e. to } z = x - \mu.$$

Entonces, la definición anterior se convierte en

$$p(X = z) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp\left(-\frac{1}{2} \frac{z^2}{\sigma^2}\right).$$

Sin embargo, [esta expresión aún no es una función de masa de probabilidad](#) sino una **función de densidad de probabilidad pdf ya que**:

- Aún no describe la probabilidad de que el RW esté en la posición  $z$ , pero la probabilidad de que un RW **con longitud de pasos en promedio** está **cerca**  $z$ .
- Como resultado,  $p(X = z) = 0$  ya que no podemos deshacer precisamente el recorrido del RW para aterrizar en un punto específico  $z$  pero **aterrizar en un intervalo**  $\Delta z$ . Cuanto más pequeño sea el  $\Delta z$  (se convierte en un diferencial  $dz \rightarrow 0$ ) más preciso es nuestro resultado (ver Conferencias Feynman), y esto sucede cuando  $n \rightarrow \infty$ :

$$p(X \in [t, t + dt]) = \sum p(z) \Delta z = \int_t^{t+dt} p(z) dz.$$

Entonces, la **distribución acumulada** queda:

$$p(X \leq t) = \int_{z \leq t} p(z) dz.$$

En particular, definamos la variable aleatoria (continua)  $Z$  donde,

$$Z = \left( \frac{x - \mu}{\sigma} \right) = \left( \frac{z}{\sigma} \right).$$

Entonces la **unidad o distribución normal estandarizada** y su **pdf** está dada por

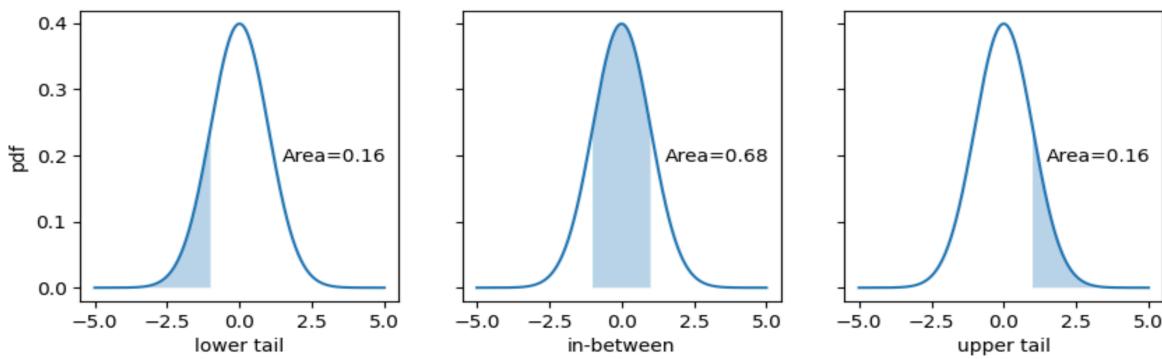
$$p(Z = z) = \phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}.$$

En realidad, la integral anterior, conocida como **integral de Gauss**, satisface los axiomas de probabilidad ya que

$$\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} dz = 1,$$

lo que significa que el **área bajo la función de Gauss** define una probabilidad. En particular, mirando [GaussU](#) tenemos:

[Saltar al contenido principal](#)

Fig. 3.6 Acumulados de la unidad gaussiana:  $a = -1, b = 1$ .

- Probabilidad de  $Z \leq a$  (cola inferior):

$$p(Z \leq a) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^a e^{-\frac{1}{2}z^2} = \phi(a).$$

- Probabilidad de  $a \leq Z \leq b$  (entre  $a$  y  $b$ ):

$$p(a \leq Z \leq b) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{1}{2}z^2} = \phi(b) - \phi(a).$$

- Probabilidad de  $Z \geq a$  (cola superior):

$$p(Z \geq a) = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^a e^{-\frac{1}{2}z^2} = 1 - \phi(a).$$

Básicamente, una vez que estandarizas una variable  $X$  (es decir, transformarlo en  $Z$ ), tienes todo lo que necesitas para calcular probabilidades gaussianas:

$$p(a \leq X \leq b) = \frac{1}{\sqrt{2\pi \cdot \sigma^2}} \int_a^b \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right) dx = \phi\left(\frac{b - \mu}{\sigma}\right) - \phi\left(\frac{a - \mu}{\sigma}\right).$$

Si desea consultar los valores de cola inferiores, visite la [tabla de probabilidad acumulativa normal estándar](#)

**Vínculo con límites fundamentales**. Una vez que hemos descubierto la naturaleza exponencial del gaussiano, y una vez que nos damos cuenta de que el gaussiano es el límite del binomio, cerramos el ciclo de comprensión de las desintegraciones exponenciales de los **límites de Hoeffding y Chernoff**.

## 3.2. Dependencia estadística

### 3.2.1. Sin reemplazo

Suponer que los eventos son iid (independientes y distribuidos de manera idéntica) está algo lejos de modelar eventos reales. La forma más sencilla de entenderlo es **cambiar las condiciones del experimento** de una prueba a otra (**sin reemplazo**).

Tomemos, por ejemplo, una baraja estándar de 52 tarjetas: 4 trajes (clubes♣, diamantes♦, espadas♠ y corazones♥) y para cada uno de ellos 2 – 10 cartas, más As, A, Jacobo J, Reina Q y rey K:  $13 \times 4 = 52$  tarjetas. Además,

- Los diamantes y los corazones son Red, mientras que los otros dos trajes son Black.
- Jotas, Reinas y Reyes son Face tarjetas.

Entonces, la probabilidad de obtener una espada es  $p(\spadesuit) = \frac{13}{52} = 1/4$ . Sin embargo:

- Si obtenemos un ♠, retíralo del mazo y vuelve a barajar, la probabilidad cambia:  $p(\spadesuit'|\spadesuit) = \frac{12}{51} = 0.235 < 1/4$ .
- También cambia si obtenemos una carta de cualquier otro palo, digamos ♦, pero de otra manera:  $p(\spadesuit'|\diamondsuit) = \frac{13}{51} = 0.254 > 1/4$ .

[Saltar al contenido principal](#)

En ambos casos, la notación  $p(A|B)$  denota la probabilidad de obtener  $A$  dado  $B$ . En ambos casos, el condicionamiento modifica la probabilidad del evento original  $A$ . Por lo tanto,  $A$  depende **condicionalmente de  $B$** . En otras palabras  $A$  y  $B$  son **independientes** sólo si

$$p(A|B) = p(A)$$

Además, la probabilidad condicional se calcula mediante el **teorema de Bayes**:

$$p(A|B) = \frac{p(A \cap B)}{P(A)}.$$

Ahora, imponga la no independencia o **el condicionamiento** (dependencia) dibujando  $k \ll 52$  tarjetas **sin reemplazo** y luego preguntando por la probabilidad de ciertos eventos relacionados con estos  $k$  tarjetas.

Por ejemplo, si sacamos dos cartas *secuencialmente*, ¿cuál es la probabilidad de obtener un as de diamantes?  $A \diamondsuit$  y un Black tarjeta .

Ya que dibujamos el  $k$  tarjetas secuencialmente, tenemos que considerar el  $k!$  posibles órdenes de obtención  $k$  tarjetas de 52. En este caso, tenemos  $k = 2$  y en consecuencia dos posibles órdenes:

- Primero  $A \diamondsuit$ , segundo Black.
- Primero Black, segundo  $A \diamondsuit$ .

Denotamos los eventos de la siguiente manera:

$$A = \{\text{First card is } A \diamondsuit\}$$

$$B = \{\text{Second card is } A \diamondsuit\}$$

$$C = \{\text{First card is Black}\}$$

$$D = \{\text{Second card is Black}\}$$

Luego, exploramos la probabilidad de cada "orden":

$$p(A \cap D) = p(A)p(D|A) = \frac{1}{52} \cdot \frac{26}{51} = \frac{1}{102}$$

$$p(C \cap B) = p(C)p(B|C) = \frac{26}{52} \cdot \frac{1}{51} = \frac{1}{102}$$

Por lo tanto, dado que ambos órdenes son disjuntos, el resultado final viene dado por

$$p(A \cap D) + p(C \cap B) = \frac{1}{102} + \frac{1}{102} = \frac{2}{102} = \frac{1}{51}.$$

**Diagramas de árbol**. Como hicimos con las monedas ( $H, T$ ) y el triángulo de Pascal (ver [HT](#)), la representación de problemas de probabilidad condicional con árboles facilita la visualización del problema y la interpretabilidad de la solución. Los árboles de probabilidad se construyen de la siguiente manera:

- La raíz del árbol es el origen del experimento.
- Todos los bordes están *dirigidos* y van de nivel  $l$  a nivel  $l+1$ . Los bordes están etiquetados con probabilidades. Los bordes que salen de un nodo dado  $n$  *Debe agregar la unidad*.
- Los bordes que salen de la raíz (nivel  $l=0$ ) están asociados con *probabilidades no condicionales de eventos*, por ejemplo  $p(A)$ .
- Los bordes en niveles  $l > 0$  están asociados con *probabilidades condicionadas* al nivel anterior, por ejemplo  $p(B|A)$ .
- Los nodos describen eventos.
- Las hojas codifican eventos interseccionales, por ejemplo  $p(A \cap C)$ .

Antes de utilizar diagramas de árbol, es interesante señalar que **los diagramas de árbol no son adecuados** para resolver problemas como el problema de la baraja de cartas anterior: ¿cuál es la probabilidad de obtener un as de diamantes?  $A \diamondsuit$  y un Black tarjeta tomada secuencialmente (sin reemplazo). ¿Por qué?

- Tomar un pedido, por ejemplo  $A, D$  (la primera carta es  $A \diamondsuit$ , el segundo es Black). Como la probabilidad de las ramas debe sumar 1, no podemos poner  $A$  y  $D$  como sucursales desde  $p(A) + p(D) \neq 1$ .
- De hecho, necesitamos un árbol para cada pedido.

[Saltar al contenido principal](#)

- En el primer árbol, codificamos el orden. $A, D$ (la primera carta es  $A \diamond$ , el segundo es Black) calcular  $p(A)yp(D|A)$  llevando  $ap(A \cap D)$ .
- En el segundo árbol, codificamos el orden. $C, B$ (la primera carta es Blacky el segundo es  $A \diamond$ ), calcular  $p(B)yp(C|B)$  llevando  $ap(C \cap B)$ .

En **Tree1**, mostramos el primero de estos dos árboles. Tenga en cuenta que la mitad de los nodos proporcionaron respuestas que no son necesarias en este problema en particular. Sin embargo, la respuesta  $ap(A \cap D)$  sumado al dep( $C \cap B$ ) dado por un segundo árbol, resuelve el problema.

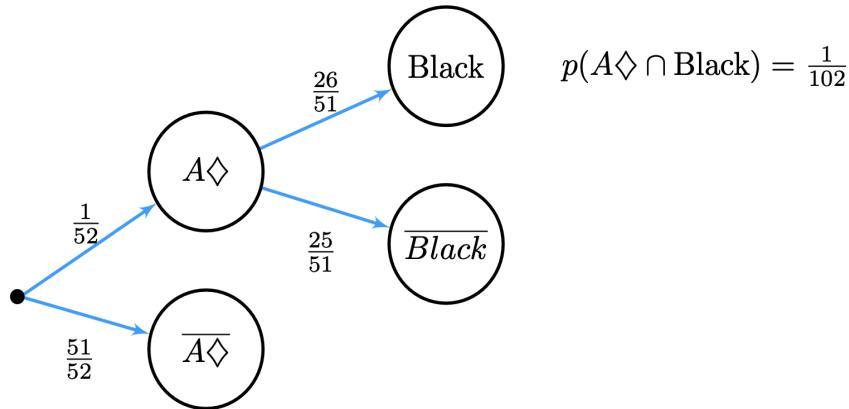


Fig. 3.7 Árbol simple para  $p(A \cap D)$ .

A pesar del ejemplo anterior, los diagramas de árbol son muy útiles en otros **problemas sin reemplazo**, como responder la siguiente pregunta sobre apuestas:

¿Cuál es la probabilidad de que después de extraer dos cartas del mazo, sin reposición, obtenga **dos cartas del mismo color**?

Construimos el diagrama de árbol de la siguiente manera:

- Nivel1: tenemos dos posibilidades  $R = \{\text{1st card is Red}\} \cup \bar{R} = \{\text{1st card is Black}\}$ .
- Nivel2: tenemos, nuevamente dos posibilidades  $R' = \{\text{2nd card is Red}\} \cup \bar{R}' = \{\text{2nd card is Black}\}$ .

Entonces, las probabilidades de las dos ramas al nivel1son:

$$p(R) = \frac{26}{52} = \frac{1}{2}$$

$$p(\bar{R}) = 1 - p(R) = 1 - \frac{1}{2} = \frac{1}{2}$$

en nivel2tenemos4sucursales:

$$p(R'|R) = \frac{26 - 1}{52 - 1} = \frac{25}{51}$$

$$p(\bar{R}'|R) = 1 - p(R'|R) = 1 - \frac{25}{51} = \frac{26}{51}$$

$$p(R'|\bar{R}) = \frac{26}{52 - 1} = \frac{26}{51}$$

$$p(\bar{R}'|\bar{R}) = 1 - p(R'|\bar{R}) = 1 - \frac{26}{51} = \frac{25}{51}$$

que conducen a4probabilidades de intersección:

$$p(R)p(R'|R) = \frac{1}{2} \cdot \frac{25}{51} = p(R \cap R')$$

$$p(R)p(\bar{R}'|R) = \frac{1}{2} \cdot \frac{26}{51} = p(R \cap \bar{R}')$$

$$p(\bar{R})p(R'|\bar{R}) = \frac{1}{2} \cdot \frac{26}{51} = p(\bar{R} \cap R')$$

$$p(\bar{R})p(\bar{R}'|\bar{R}) = \frac{1}{2} \cdot \frac{25}{51} = p(\bar{R} \cap \bar{R}')$$

[Saltar al contenido principal](#)

$$p(R \cap R') + p(\bar{R} \cap \bar{R}') = \frac{1}{2} \cdot \frac{25}{51} + \frac{1}{2} \cdot \frac{25}{51} = \frac{25}{51} .$$

Básicamente, cuando extraemos una carta de un color determinado, reducimos **las probabilidades** de extraer una muestra del mismo color en el futuro y **aumentamos** las del color opuesto. En otras palabras, estamos preguntando la probabilidad de **desequilibrar las cuotas**. La probabilidad de equilibrar las probabilidades es en realidad

$$p(R \cap \bar{R}') + p(\bar{R} \cap R') = 1 - \left( p(R \cap R') + p(\bar{R} \cap \bar{R}') \right) = \frac{26}{51},$$

es decir, ligeramente superior al de desequilibrar las probabilidades.

Mostramos el diagrama de árbol en [Tree2](#). Tenga en cuenta que en cada nivel del árbol tenemos que la probabilidad de todas las hojas suma uno.

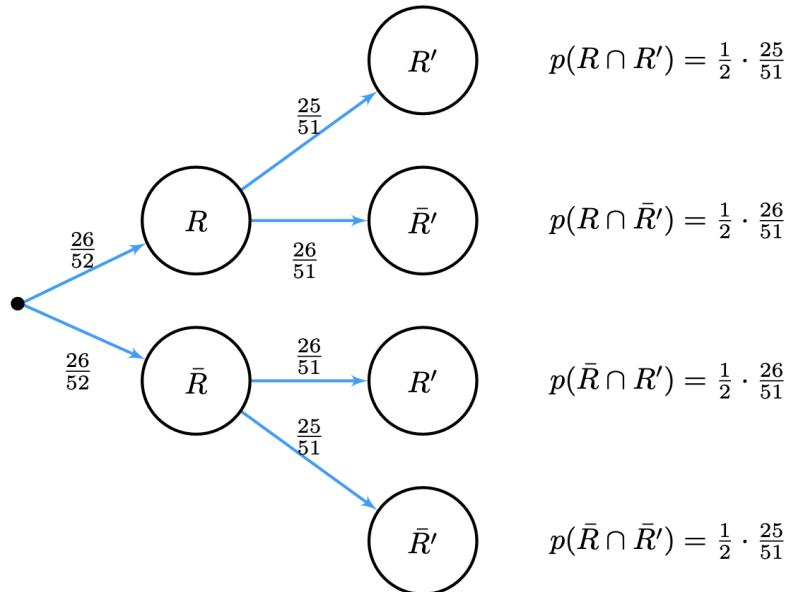


Fig. 3.8 Diagrama de árbol de una baraja de cartas (sin reemplazo).

Tenga en cuenta también que  $p(R \cap \bar{R}') = p(\bar{R} \cap R')$  y la segunda y tercera hojas se pueden fusionar en una sola con probabilidad  $\frac{1}{2} \cdot \frac{26}{51} = \frac{26}{51}$ . En realidad, ambos eventos representan el mismo resultado en diferente orden si se considera que obtener una tarjeta roja es un "éxito" (ya sea en la primera o en la segunda vuelta) y no obtenerla un "fracaso".

Por lo tanto, en cierto sentido, este tipo de árbol nos recuerda al árbol de Pascal, pero **esta vez describiendo eventos o variables condicionales** en lugar de independientes.

### 3.2.2. Expectativas condicionales

Antes de profundizar en los “árboles condicionales”, es interesante redefinir las expectativas en términos de probabilidades condicionales.

Considere, por ejemplo, **Tree3** dónde *generalizamos* el árbol de **Tree2** la siguiente manera.

- Los **nodos** se consideran como los *estados* de un sistema aleatorio con una tupla.(cards, deck), dónde cards indica el número de cartas rojas que quedan en la baraja, y deck es el número de cartas (tanto rojas como negras) que quedan en la baraja.
  - Los **bordes** están etiquetados con la probabilidad condicional de llegar al nodo de destino desde el original. Las probabilidades queemanan de un mismo nodo deben sumar uno.

Ahora, definimos las siguientes variables **aleatorias**:

- $X_0 = \frac{a}{A}$  es la fracción de cartas rojas de la baraja original donde:
    - Ason las cartas numéricas del mazo original (52).
    - aes el número de cartas rojas en la baraja (26).

[Saltar al contenido principal](#)

- $X_2$  es la fracción de tarjetas rojas que quedan después del segundo sorteo sin **reemplazo**.

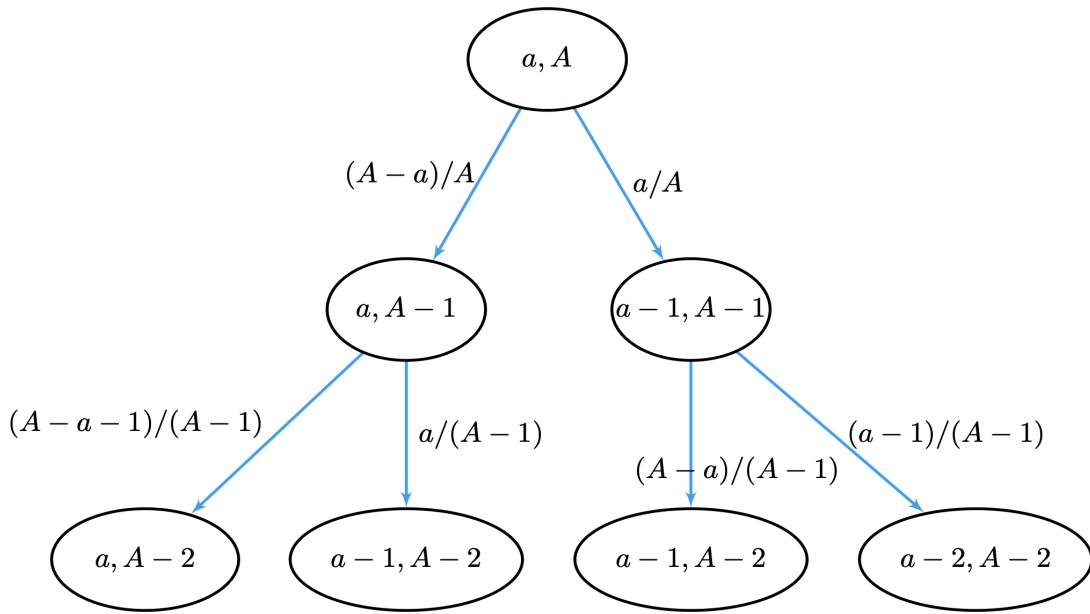


Fig. 3.9 Diagrama de árbol general para una baraja de cartas (sin reemplazo).

Luego, definimos la **expectativa condicional**  $E(X|Y = y)$  de  $X$  wrt a la configuración  $Y = y$  como sigue:

$$E(X|Y = y) = \sum_x x \cdot p(X = x|Y = y) = \sum_x x \cdot \frac{p(X = x, Y = y)}{p(Y = y)},$$

dónde  $p(X = x, Y = y)$  es la **probabilidad conjunta** (intersección).

Considere, por ejemplo,  $X = X_1$  y  $Y = X_0$ . Desde  $X_1$  tiene dos estados, comenzamos definiendo las probabilidades condicionales:

$$\begin{aligned} p\left(X_1 = \frac{a}{A-1} \mid X_0 = \frac{a}{A}\right) &= \frac{A-a}{A} = p\left(X_1 = \frac{a}{A-1}\right) \\ p\left(X_1 = \frac{a-1}{A-1} \mid X_0 = \frac{a}{A}\right) &= \frac{a}{A} = p\left(X_1 = \frac{a-1}{A-1}\right) \end{aligned}$$

es decir, saber  $X_0 = \frac{a}{A}$  no modifica la probabilidad de  $X_1$  ( $X_0$  y  $X_1$  son independientes).

Desde  $X_0$  tiene un solo valor, tenemos

$$\begin{aligned} E(X_1|X_0) &= \sum_x x \cdot p\left(X_1 = x \mid \frac{a}{A}\right) \\ &= \frac{a}{A-1} \cdot \frac{A-a}{A} + \frac{a-1}{A-1} \cdot \frac{a}{A} \\ &= \frac{a}{A} \left[ \frac{(A-a) + (a-1)}{(A-1)} \right] \\ &= \frac{a}{A} \left[ \frac{A-1}{A-1} \right] \\ &= \frac{a}{A} \end{aligned}$$

Desde  $X_0$  y  $X_1$  son independientes, tenemos eso  $E(X_1|X_0) = E(X_1)$ .

Sin embargo  $X_2$  depende claramente de  $X_1$ . mirando el nivel  $l = 2$  tenemos 3 diferentes valores para  $X_2$ :  $\frac{a}{A-2}$ ,  $\frac{a-1}{A-2}$  y  $\frac{a-2}{A-2}$ .

[Saltar al contenido principal](#)

Calculemos las probabilidades condicionales para  $X_2$  (en realidad, la probabilidad de cada hoja del árbol). Aquí aplicamos la **regla de la cadena** para probabilidades condicionales  $p(X|Y, Z) = p(X|Y)p(Y|Z)$ .

$$\begin{aligned} p(X_2|X_1, X_0) &= p(X_2|X_1)p(X_1|X_0) \\ &= p(X_2 = x_2|X_1 = x_1)p\left(X_1 = x_1 | X_0 = \frac{a}{A}\right) \end{aligned}$$

Luego, mirando el árbol, consideramos los caminos que conducen a cada una de sus diferentes hojas:

- $X_2 = 0$  red cards extracted (rama izquierda):

$$p\left(X_2 = \frac{a}{A-2} | X_1 = \frac{a}{A-1}\right) \frac{A-a}{A} = \frac{A-a-1}{A-1} \cdot \frac{A-a}{A}$$

- $X_2 = 1$  red card extracted (ramas intermedias):

$$\begin{aligned} p\left(X_2 = \frac{a-1}{A-2} | X_1 = \frac{a}{A-1}\right) \frac{A-a}{A} + p\left(X_2 = \frac{a-1}{A-2} | X_1 = \frac{a-1}{A-1}\right) \frac{a}{A} = \\ \frac{a}{A-1} \cdot \frac{A-a}{A} + \frac{A-a}{A-1} \cdot \frac{a}{A} = 2 \frac{A-a}{A-1} \cdot \frac{a}{A} \end{aligned}$$

- $X_2 = 2$  red cards extracted (rama derecha):

$$p\left(X_2 = \frac{a-2}{A-2} | X_1 = \frac{a-1}{A-1}\right) \frac{a}{A} = \frac{a-1}{A-1} \cdot \frac{a}{A}$$

Luego procedemos a calcular  $E(X_2|X_1, X_0)$  como sigue:

$$\begin{aligned} E(X_2|X_1, X_0) &= \sum_{x_2} x_2 \cdot p(X_2 = x_2|X_1, X_0) \\ &= \frac{a}{A-2} \cdot \frac{A-a-1}{A-1} \cdot \frac{A-a}{A} + \frac{a-1}{A-2} \cdot 2 \frac{A-a}{A-1} \cdot \frac{a}{A} + \frac{a-2}{A-2} \cdot \frac{a-1}{A-1} \cdot \frac{a}{A} \end{aligned}$$

Mire **atentamente** el patrón de la expresión anterior :

- nos movemos de 0 éxitos (tarjeta roja extraída) para 1 y 2 éxitos.
- Cada éxito se caracteriza por  $a - k$ , con  $k = 0, 1, 2$ .
- Cada fracaso se caracteriza por  $A - a - l$ , con  $l = 0, 1$ .

Reorganizando adecuadamente cada término para que las fallas aparezcan primero, tenemos

$$\begin{aligned} E(X_2|X_1, X_0) &= \frac{A-a}{A-2} \cdot \frac{A-a-1}{A-1} \cdot \frac{a}{A} + 2 \frac{A-a}{A-2} \cdot \frac{a-1}{A-1} \cdot \frac{a}{A} + \frac{a-2}{A-2} \cdot \frac{a-1}{A-1} \cdot \frac{a}{A} \\ &= \frac{a}{A} \left[ \frac{A-a}{A-2} \cdot \frac{A-a-1}{A-1} + 2 \frac{A-a}{A-2} \cdot \frac{a-1}{A-1} + \frac{a-2}{A-2} \cdot \frac{a-1}{A-1} \right] \\ &= \frac{a}{A} \left[ \frac{A^2 - 3A + 2}{(A-1)(A-2)} \right] \\ &= \frac{a}{A} \left[ \frac{(A-1)(A-2)}{(A-1)(A-2)} \right] \\ &= \frac{a}{A} \end{aligned}$$

Por lo tanto, tenemos que  $E(X_2|X_1, X_0) = E(X_1)$ . Esta propiedad no se satisface, por lo general, con ninguna expectativa condicional, pero cuando sucede decimos que tenemos una **martingala**.

### 3.2.3. Martingalas

Dada una secuencia de variables aleatorias  $X_0, X_1, \dots, X_n, X_{n+1}$ , es una martingala si

[Saltar al contenido principal](#)

$$E(X_{n+1}|X_n, \dots, X_1, X_0) = E(X_n) \text{ for all } n \geq 0.$$

En el ejemplo anterior,  $E(X_{n+1}|X_n, \dots, X_1, X_0) = \frac{a}{A} = E(X_1)$ , incluso si las variables están condicionadas.

### ¿Por qué las martingalas son útiles en la Inteligencia Artificial?

Las martingalas son **procesos aleatorios o estocásticos** no tan simples que las sumas de iids (lanzamiento de monedas) pero no demasiado complejos para estudiar. Curiosamente, **los paseos aleatorios** son casos particulares de martingalas.

La idea detrás de las martingalas es que **las expectativas nunca cambian** incluso cuando agregas un nuevo nivel en el árbol (una nueva variable condicionada). En promedio, el valor de la variable  $X_{n+1}$  es el de  $E(X_n)$  lo cual *no significa* que  $X_{n+1}$  sólo está condicionado a  $X_n$  como ocurre en las cadenas de Markov. De hecho,  $X_{n+1}$  está condicionado a  $X_n, X_{n-1}, \dots, X_0$  pero la expectativa condicional es **invariante**.

**Juegos justos**. La invariancia de la expectativa condicional explica la aplicación de Martingalas para modelar las expectativas de los jugadores en juegos limpios:

Definamos las apuestas de un jugador. 1 moneda para el Casino sacando una carta roja de la baraja. Si ganamos, él obtiene 1 atrás. Al hacerlo, el beneficio esperado es **constante**. ¿Por qué?

- $X_n$  modela la fortuna del jugador al final del  $n^{th}$  jugar.
- Si el juego es justo, la **fortuna esperada**  $E(X_{n+1})$  en el juego  $n+1$  es el mismo que el del juego  $n$   $E(X_n)$ , es decir, la información condicional no puede predecir el futuro.

### 3.2.4. Vínculos con el Triángulo de Pascal

Observando atentamente la estructura de  $E(X_2|X_1, X_0)$  tenemos que es igual a:

$$\frac{a}{A-2} \cdot \underbrace{\binom{2}{0} \frac{A-a-1}{A-1} \cdot \frac{A-a}{A}}_{p(R_2=0)} + \frac{a-1}{A-2} \cdot \underbrace{\binom{2}{1} \frac{A-a}{A-1} \cdot \frac{a}{A}}_{p(R_2=1)} + \frac{a-2}{A-2} \cdot \underbrace{\binom{2}{2} \frac{a-1}{A-1} \cdot \frac{a}{A}}_{p(R_2=2)}$$

dónde  $p(R_2 = k)$  es la probabilidad de sacar  $k$  tarjetas rojas al nivel  $n = 2$ .

En general tenemos:

$$p(R_n = k) = \binom{n}{k} \frac{P(A-a, n-k) \cdot P(a, n)}{P(A, n)}$$

dónde  $P(n, r) = n(n-1) \dots (n-r+1)$  es una **r-permutación** deno como se define en el tema de combinatoria. La expresión anterior surge de observar los patrones factoriales tanto en el numerador como en el denominador.

Comparado con el caso iid (**Bern**), es decir, con reemplazo, donde la probabilidad de obtener  $k$  las tarjetas rojas deben ser binomiales

$$p(R_n = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

con  $p = 1/2$ , en **PascalMartín** mostramos, con colores, la distribución de probabilidad para el caso condicional, es decir, para la martingala, donde mantuvimos  $\frac{a}{A} = p$  por ser comparable al caso independiente.

[Saltar al contenido principal](#)

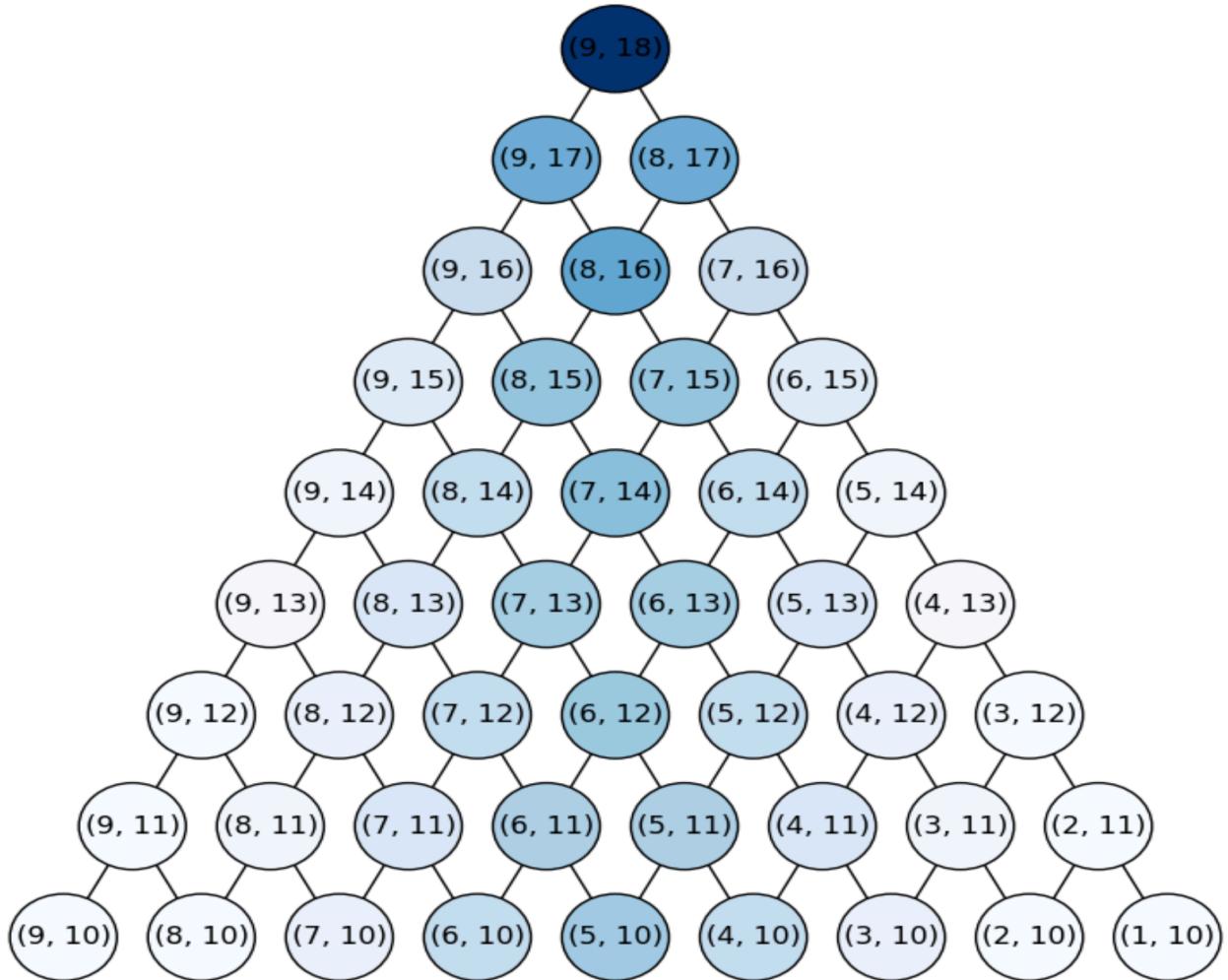


Fig. 3.10 Distribución de una martingala.

Tenga en cuenta que:

- Los eventos extremos (todos los fracasos/todos los éxitos) tienden a tener una probabilidad cero comoncrece.
- La mayor parte de la distribución está cerca de  $E(X_1) = \frac{a}{A}$  pero comonaumenta la pmf (función puntual-masa) se aplana.
- Aplanamiento connse debe al denominador  $P(A, n)$  dep( $R_n = k$ ).
- Por supuesto, podemos adaptar las igualdades fundamentales definidas para las variables iid a las condicionales, pero está bastante claro que los eventos raros serán menos probables en árboles condicionales como el de **PascalMartin** a menos que cambiemos la relación entre  $A$ .

< Anterior

[2. Combinatoria como conteo](#)

Próximo >

[4. Paseos aleatorios sobre gráficos.](#)

# Paseos aleatorios en gráficos.

## Contenido

- 4.1. cadenas de markov
- 4.2. Relaciones de recurrencia
- 4.3. Paseos aleatorios en 2D
- 4.4. El fenómeno del límite

### 4.1. Cadenas de Markov

Hasta ahora, hemos estudiado tanto **procesos aleatorios independientes** como **procesos aleatorios condicionales**. En este sentido, cuando tenemos independencia, nuestro proceso aleatorio es un *paseo aleatorio simple* (ver [RWrand](#)). Sin embargo, cuando las variables del proceso aleatorio están completamente condicionadas, el proceso aleatorio correspondiente es más difícil de estudiar a menos que tengamos una martingala. Afortunadamente, [hay algo a medio camino entre la sencillez de los procesos aleatorios independientes y el pleno condicionamiento de la martingala](#): nos referimos a **las cadenas de Markov**.

**Cadena de Markov**. Una secuencia de variables aleatorias  $X_0, X_1, \dots$  es una cadena de Markov si para todos los estados posibles (valores de las variables aleatorias)  $x_{t+1}, x_t, \dots, x_1$  tenemos:

$$p(X_{t+1} = j | X_t = i, \dots, X_1 = x_1, X_0 = x_0) = p(X_{t+1} = j | X_t = i) = p_{ij}$$

with  $p_{ij} \geq 0 \forall i, j$  and  $\sum_j p_{ij} = 1 \forall i$ ,

es decir, la probabilidad de un evento futuro sólo depende de la de uno presente. Esto se llama **propiedad de Markov** o **propiedad sin memoria**.

Un par de propiedades interesantes:

- **Irreductibilidad**. Decimos que un estado  $j$  en la cadena de Markov (MC) es **accesible** desde otro estado  $i$  si existe  $t \geq 0$  de modo que

$$p(X_t = j | X_0 = i) = p_{ij}(t) > 0 \text{ in } t \text{ steps ,}$$

es decir, podemos obtener de un estado  $i$  a otro  $j$  en  $t$  pasos con probabilidad  $p_{ij}(t)$ . Entonces, un MC es **irreducible** si cada par  $(i, j)$  de los estados es mutuamente accesible.

- **Periodicidad**. Un estado  $i$  tiene **periodo**  $d_i$  si

$$d_i = \gcd(t \in \{1, 2, \dots\} : p_{ii}(t) > 0)$$

dónde  $\gcd$  denota el máximo común divisor. Entonces si  $d_i > 1$  el estado  $i$  es **periódico**; si  $d_i = 1$  es **aperiódico**. Entonces un MC es **aperiódico** si todos los estados tienen período  $d_i = 1$ .

**Gráficos**. Un gráfico  $G = (V, E)$  consiste en un conjunto de **nodos** o vértices  $V = 1, 2, \dots, n$  donde  $|V| = n$  y un conjunto de **aristas**  $E \subseteq V \times V$ .

- Un borde  $e = (i, j)$  se denota por un par de nodos  $i$  y  $j$ , donde  $i$  es el origen y  $j$  el objetivo o destino.
- Si el gráfico **no** está dirigido tanto  $(i, j)$  y  $(j, i)$  existen para todo  $e \in E$ . En caso contrario, la gráfica está **dirigida**.
- Si  $e = (i, i)$  tenemos un **auto-bucle**.

Los gráficos son herramientas matemáticas muy flexibles. Comenzamos usándolos para describir un proceso aleatorio. Los nodos  $V$  proporcionan los **estados** y los bordes  $E$  proporcionan las **transiciones entre estados**. En realidad, etiquetamos los bordes con la probabilidad de una transición markoviana  $p(j|i) = p_{ij}$ .

El siguiente ejemplo está motivado por el ensayo [Random Walks and Electric Networks](#) de Doyle y Snell.

[Saltar al contenido principal](#)

Queremos construir un gráfico para el **paseo del borracho**. Un hombre camina por un  $\mathbb{Z}$ —Las cuadras se extienden en Madison Avenue. Él comienza en la esquina  $(x, y)$  con probabilidad  $1/2$  camina una cuadra hacia la derecha  $y$ , también con probabilidad  $1/2$  camina una cuadra hacia la izquierda. En la siguiente esquina, vuelve a elegir su dirección al azar. Continúa hasta llegar a la esquina  $(5, 5)$ , que es casa o esquina  $0$ , que es un bar. En ambos casos permanece allí.

Nuestro gráfico para esta caminata tienen  $= 6$  vértices o **estados**  $V = \{0, 1, 2, 3, 4, 5\}$ , donde  $0$  es Homey y  $5$  es el Bar. Vea **Drunk** dónde están los bordes o **transiciones** en azul (bidireccionales si no representamos las puntas de flecha) y la posible decisión para el nodo  $x = 3$  están representados en negro.

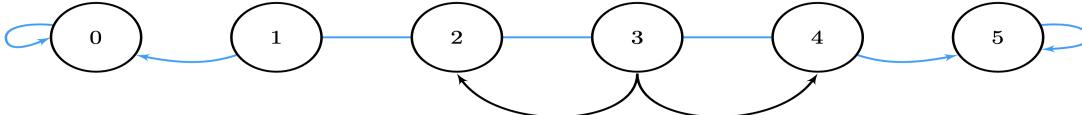


Fig. 4.1 Gráfico del paseo del borracho.

En realidad, los bordes del gráfico anterior son  $E = \{(0, 0), (1, 0), (1, 2), (2, 1), \dots, (4, 5), (5, 5)\}$

¿Por qué nuestro gráfico imita el **andar del borracho** y por qué es un **MC**?

- Tenemos dos estados,  $0$  y  $1$ , con **bucles propios** y sin bordes para regresar a cualquier otro nodo. Estos estados se denominan **estados absorbentes** en la terminología MC, ya que una vez que la caminata aleatoria en cadena de Markov (MCRW) los alcanza, queda atrapada allí. Como resultado, el CM es **reducible**.
- El gráfico es casi **bipartito**, es decir, podemos dividir  $V$  en subconjuntos  $V_1 = \{0, 2, 4\}$  y  $V_2 = \{1, 3, 5\}$  para que los nodos en  $V_1$  sólo pueden ir nodos de  $V_2$  (excepto los nodos absorbentes) y viceversa. Como resultado, el CM es casi **aperiodico**. Los nodos no absorbentes tienen período  $2$  y los absorbentes tienen punto  $1$ .
- Si iniciamos nuestro MCRW en un estado no absorbente, digamos  $x \in \{2, 3, 4\}$  caminamos hacia la izquierda con probabilidad  $1/2$  y a la derecha también con probabilidad  $1/2$ .
- El **grado**  $\deg(i)$  de un nodo  $i$  en la gráfica está el número de vecinos  $\mathcal{N}_i$ , es decir, el número de **bordes salientes** de  $i$ . En nuestras gráficas tenemos que todos los nodos no absorbentes tienen grado  $2$  mientras que los estados absorbentes tienen grado  $1$  (en realidad sus vecinos son ellos mismos).
- El  $\text{gradodeg}(i)$  de cada nodo  $i$  revela que la probabilidad (Markoviana) de hacer una transición a un vecino es

$$p(j|i) = p_{ij} = \frac{a_{ij}}{\deg(i)} \quad \text{where } a_{ij} = \begin{cases} 1 & \text{if } i \in \mathcal{N}_j \\ 0 & \text{otherwise.} \end{cases}$$

ya  $a_{ij} = 1$  significa que los nodos  $i$  y  $j$  son **adyacentes**. Para los estados absorbentes, tenemos  $a_{ij} = 0 \forall j \neq i$ . Como resultado, para estos estados  $p_{ii} = 1$  y  $p_{ij} = 0 \forall j \neq i$ .

**Patrones del andar del borracho**. Para tener una idea aproximada del comportamiento de este proceso de Markov, hemos generado 40,000 paseos aleatorios (10,000 comenzando en cada estado no absorbente  $x = 1, 2, 3, 4$ ). Cada paseo aleatorio tiene una longitud  $l = 10$ . ¿Por qué? Lo descubriremos en breve. Lo importante ahora es señalar que algunos de los paseos terminan en uno de los estados absorbentes ( $x = 0$ ), mientras que muchos otros terminan en el otro ( $x = 5$ ). La probabilidad de alcanzar cada estado absorbente es  $1/2$ .

Trazamos estos paseos **Drunkard**. Cuanto más oscura es la línea azul, más *lento* llega el paseo  $x = 5$ . Esto significa que si las caminatas llegan  $x = 5$  en la longitud máxima del camino  $l = 10$  se vuelve el más oscuro.

Algunos patrones interesantes:

- Como comenzamos uniformemente el mismo número de caminos en cada estado no absorbente, no hay una diferencia clara entre los caminos que terminan en  $x = 0$  y los que terminan en  $x = 5$ .
- Además, algunos caminos que tienden a  $x = 0$  girar repentinamente hacia  $x = 5$  y viceversa.

Aparentemente todos los estados no absorbentes alcanzan  $x = 5$  igualmente lentamente. **Sin embargo, esto es engañoso**. En realidad, la mayoría de los caminos llegan  $x = 5$  muy temprano. Esto sugiere que la probabilidad de alcanzar  $x = 5$  desde cualquier estado no absorbente *no es uniforme*. Esto nos lleva a la respuesta a la primera pregunta a resolver sobre un paseo

[Saltar al contenido principal](#)

## 4.2. Relaciones de recurrencia

Antes de abordar las **dos preguntas fundamentales** para un MCRW: (a) hacia dónde converge y (b) cuánto tiempo lleva, es importante señalar que el resto de esta sección requiere algo de práctica con solucionadores algebraicos de relaciones recurrentes. a saber, **ecuaciones en diferencias lineales**. Son herramientas muy prácticas que están explicadas en el excelente [Github de Mathew Aldridge](#) e incluso en la [Wikipedia](#). Los siguientes ejemplos han sido adaptados de la primera fuente y resuelven algunos de los ejercicios de [Random Walks and Electric Networks](#) de Doyle y Snell.

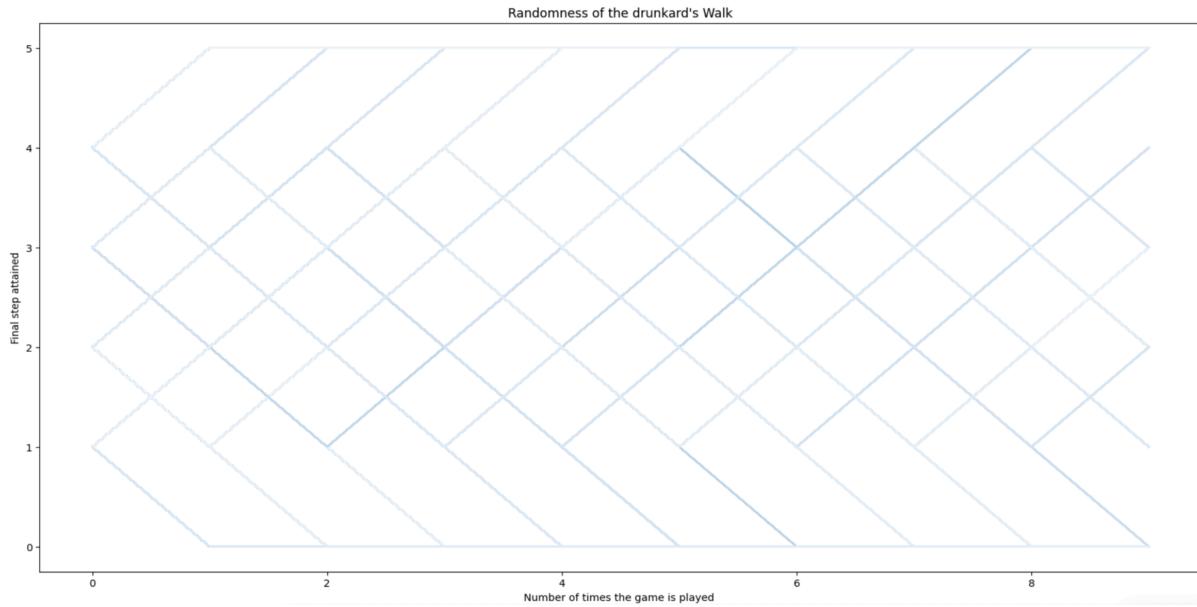


Fig. 4.2 Patrones de paseos de borrachos generados.

**P1.** ¿Cuál es la probabilidad de terminar en **Home** si partimos del estado no absorbente  $x$ ?

En otras palabras, ¿cuál es la probabilidad de **acertar**?  $x = 5$  de **antes de golpear**  $x = 0$ ?

1. Comenzamos **formulando la markovianidad** del camino del borracho de una manera más genérica:

$$x_{n+1} = \begin{cases} x_n + 1 & \text{with probability } p \text{ if } 1 \leq n \leq m-1 \\ x_n - 1 & \text{with probability } q \text{ if } 1 \leq n \leq m-1 \\ 0 & \text{if } n = 0 \\ m & \text{if } n = m, \end{cases}$$

dónde:  $x_n$  es la posición del paso a  $n$ ,  $p = 1 - q = 1/2$  es la probabilidad de una transición desde un estado no absorbente y  $m = 5$  es el nodo objetivo.

2. Planteamos la fórmula anterior en términos probabilísticos usando la **condición del primer paso**:

$$\begin{aligned} p(\text{Home}) &= p(\text{1st step right})p(\text{Home}|\text{1st step right}) \\ &\quad + p(\text{1st step left})p(\text{Home}|\text{1st step left}) \\ &= p \cdot p(\text{Home}|\text{1st step right}) + q \cdot p(\text{Home}|\text{1st step left}). \end{aligned}$$

Aquí utilizamos el **teorema de probabilidad total** para los siguientes eventos:

$$\begin{aligned} E &= \{\text{Home}\} \\ A &= \{\text{1st step right}\} \\ \bar{A} &= \{\text{1st step left}\} \end{aligned}$$

Entonces, probabilidad total significa que la probabilidad de un evento dados dos (o más) eventos exclusivos es

[Saltar al contenido principal](#)

$$p(E) = p(E \cap A) + p(E \cap \bar{A}) = p(A)p(E|A) + p(\bar{A})p(E|\bar{A}) .$$

En nuestro caso:

$$P(E) = p \cdot p(E|A) + q \cdot p(E|\bar{A}) ,$$

y queremos calcular ambos  $p(E|A)$  y  $p(E|\bar{A})$  para calcular  $P(E)$ .

### 3. Formule y resuelva una **relación de recurrencia**:

Entonces, tenemos que resolver la siguiente relación de recurrencia:

$$r_n = p \cdot r_{n+1} + q \cdot r_{n-1} \text{ subject to } r_0 = 0, r_m = 1 .$$

dónde  $r_0 = 0$  y  $r_m = 1$  son las **condiciones límite** que especifican el éxito si alcanzamos  $m$  (Home) y el fracaso si llegamos 0 (Bar).

Formulamos la relación de recurrencia como una **ecuación en diferencias lineal**. En este caso es **homogéneo** (como un sistema lineal homogéneo  $\mathbf{Ax} = \mathbf{0}$ ) desde:

$$p \cdot r_{n+1} + q \cdot r_{n-1} - r_n = 0 .$$

En primer lugar aplicamos el siguiente cambio de variable:

$$r_n = \lambda^n ,$$

lo que lleva a

$$p\lambda^{n+1} + q\lambda^{n-1} - \lambda^n = 0 ,$$

y tomando  $\lambda^{n-1}$  como factor común produce

$$\lambda^{n-1}(p\lambda^2 + q - \lambda) = 0$$

dónde  $p\lambda^2 + q - \lambda = 0$  es la **ecuación característica** de la recurrencia. Luego, reorganizando los coeficientes tenemos  $p\lambda^2 - \lambda + q = 0$ . Para resolver esta ecuación cuadrática es conveniente utilizar el teorema del resto (Ruffini). Los divisores del término independiente ( $q$ ) son  $\pm 1$  y  $\pm q$ . Si lo intentamos primero  $+1$  y aplicar  $q = 1 - p$ , esto lleva a la ecuación  $\lambda p - q = 0$  con rendimientos  $\lambda = \frac{q}{p}$  que llamamos  $\rho$ . Entonces, la factorización que estamos buscando es

$$(p\lambda - q)(\lambda - 1) \text{ and roots } \lambda_1 = 1, \lambda_2 = \rho .$$

Ahora tenemos dos casos:

- Si  $\rho \neq 1$ , entonces tenemos dos raíces distintas. En este caso, la solución general de una ecuación homogénea tiene la forma:

$$r_n = A\lambda_1^n + B\lambda_2^n = A1^n + B\rho^n = A + B\rho^n .$$

Para determinar  $A$  y  $B$  explotamos las dos **condiciones de contorno**  $r_0 = 0, r_m = 1$ :

$$\begin{aligned} r_0 &= A + B\rho^0 = A + B = 0 \\ r_m &= A + B\rho^m = 1 \end{aligned}$$

De  $A + B = 0$  obtenemos  $A = -B$  lo que lleva a

[Saltar al contenido principal](#)

$$\begin{aligned} -B + B\rho^m &= 1 \Rightarrow (\rho^m - 1)B = 1 \Rightarrow B = \frac{1}{\rho^m - 1} \\ A &= -B = -\frac{1}{\rho^m - 1}. \end{aligned}$$

y como un resultado

$$\begin{aligned} r_n &= A\lambda_1^n + B\lambda_2^n \\ &= -\frac{1}{\rho^m - 1} + \frac{\rho^n}{\rho^m - 1} \\ &= \frac{\rho^n - 1}{\rho^m - 1} \end{aligned}$$

- Si  $\rho = 1$  tenemos  $p = q$  esto significa que tenemos dos soluciones repetidas  $\lambda_1 = \lambda_2 = 1$ . En este caso, la solución general tiene la siguiente forma:

$$r_n = (A + nB)\lambda_1^n,$$

lo que lleva a

$$\begin{aligned} r_0 &= (A + 0B)1^0 = A = 0 \\ r_m &= (A + mB)1^m = A + mB = 1 \end{aligned}$$

cuyas soluciones son  $A = 0$  y  $B = \frac{1}{m}$ .

Finalmente

$$\begin{aligned} r_n &= (A + nB)\lambda_1^n \\ &= (0 + n\frac{1}{m}) \\ &= \frac{n}{m} \end{aligned}$$

El resultado genérico es:

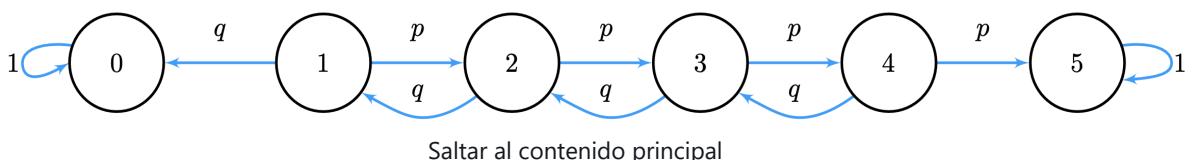
$$r_n = \begin{cases} \frac{\rho^n - 1}{\rho^m - 1} & \text{if } p \neq q \\ \frac{n}{m} & \text{if } p = q \end{cases} \quad \text{i.e.} \quad r_n = \begin{cases} \frac{\left(\frac{q}{p}\right)^n - 1}{\left(\frac{q}{p}\right)^m - 1} & \text{if } p \neq q \\ \frac{n}{m} & \text{if } p = q \end{cases}$$

**Resultado de la caminata ubiased**. Si  $p = q = 1/2$  tenemos un proceso aleatorio según el gráfico de [Drunkard](#). La probabilidad de conseguir Home, es decir, de golpear  $x = m = 5$  antes de golpear  $x = 0$  es  $p(x) = \frac{x}{m} = \frac{x}{5}$ . Cuanto más cerca estemos de Home lo más probable es que lleguemos allí. Tenga en cuenta que si invertimos las condiciones de contorno al ceder yendo a la Bar, la probabilidad de llegar allí antes de llegar a casa es  $p(x) = 1 - \frac{x}{5}$ .

Sin embargo, como  $m \rightarrow \infty$  la probabilidad de conseguir Home tiende a cero, es decir  $p(x) \rightarrow 0$ .

El resultado para el paseo imparcial (justo) es consistente con nuestras observaciones en [Drunkard](#) y muestra que la probabilidad de alcanzar Home es **uniforme**. Este resultado también explica por qué la mayoría de los 40,000 paseos aleatorios lanzados uniformemente desde cualquiera de los estados no absorbentes alcanzan Home muy pronto.

**Resultado del paseo sesgado**. Sin embargo, los paseos sesgados se modelan de otra manera. Etiquetamos las aristas con sus probabilidades. Luego, en lugar de obtener las probabilidades de transición del grado, simplemente establecemos  $p_{ij} = a_{ij}p$  o  $p_{ij} = a_{ij}q$  como en [Drunkpq](#)



[Saltar al contenido principal](#)

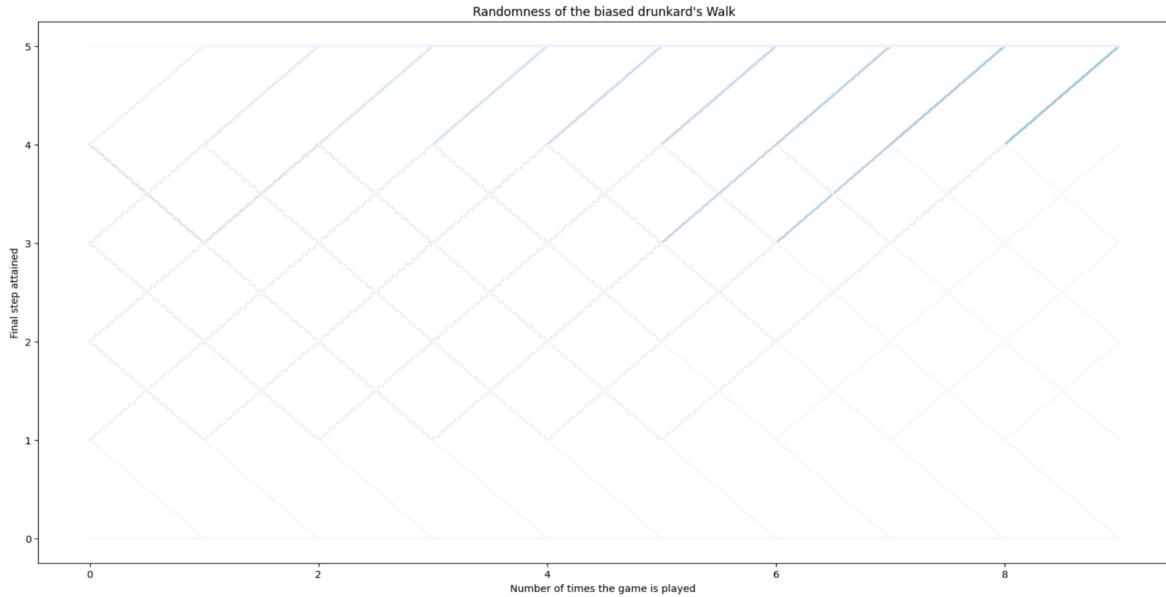
Fig. 4.3 Gráfico de paseos de borrachos sesgados.

Si  $q \ll p$ , entonces  $p(x) \rightarrow 1$  ya que estamos desviados hacia la derecha. Simétricamente, si  $q \gg p$ , entonces  $p(x) \rightarrow 0$  desde  $m > n$ .

Como  $m \rightarrow \infty$ , tenemos  $\text{esp}(x) \rightarrow 0$ , independientemente de la relación entre  $p$  y  $q$  desde

$$\lim_{m \rightarrow \infty} \frac{\rho^n - 1}{\rho^m - 1} = \lim_{m \rightarrow \infty} \frac{\rho^n / \rho^m - 1 / \rho^m}{1 - 1 / \rho^m} = \lim_{m \rightarrow \infty} \frac{0 - 0}{1 - 0} = 0 .$$

En [Drunkardpq](#) donde  $p = 0.25, q = 0.75$  podemos ver que pocos paseos llegan  $x = 5$ , en realidad la proporción de caminos "exitosos"  $\text{esp}$ .

Fig. 4.4 Patrones de paseos de borrachos sesgados generados con  $p = 0.25$ .

**P2.** ¿Cuál es la hora prevista o **hora** de llegada? Homenos partimos del estado no absorbente  $x$ ?

Estamos interesados en estimar el **tiempo esperado de impacto** de  $x = 5$ .

Como antes, nos basamos en ecuaciones en diferencias lineales.

1. Planteamos la fórmula anterior en términos probabilísticos usando la **condición del primer paso**:

$$\begin{aligned} E(\text{Duration}) &= p(1\text{st step right})p(\text{Duration}|1\text{st step right}) \\ &\quad + p(1\text{st step left})p(\text{Duration}|1\text{st step left}) \\ &= p \cdot p(\text{Duration}|1\text{st step right}) + q \cdot p(\text{Duration}|1\text{st step left}) . \end{aligned}$$

Aquí utilizamos las **expectativas condicionales** para las siguientes variables aleatorias:

$$\begin{aligned} X &= \{\text{Duration}\} \\ Y &= \{1\text{st step}\} \end{aligned}$$

dónde  $Y$  tiene valores  $y = \text{left}$  y  $y = \text{right}$ .

Entonces,

$$E(X|Y = y) = \sum_x x P(X = x|Y = y) .$$

[Saltar al contenido principal](#)

$$E(X) = E(E(X|Y)) = \sum_y p(Y=y)E(X|Y=y) .$$

dónde  $Y$  es el resultado del primer paso.

A este respecto,

$$\begin{aligned} E(X|Y=\text{left}) &= 1 + d_{n-1} \\ E(X|Y=\text{right}) &= 1 + d_{n+1} . \end{aligned}$$

Siempre cuenta 1 porque habíamos dado un paso.

Esto nos lleva a la siguiente **relación de recurrencia no homogénea**:

$$d_n = p(1 + d_{n+1}) + q(1 + d_{n-1}) = 1 + pd_{n+1} + qd_{n-1} .$$

y tenemos

$$pd_{n+1} - d_n + qd_{n-1} = -1 \text{ subject to } d_0 = 0, d_m = 0 .$$

Cuyo tamaño izquierdo lhs conduce a la misma ecuación de recurrencia homogénea que hemos estudiado antes.

- Si  $p \neq 1$  la solución general ( $\lambda_1 = 1, \lambda_2 = \rho$  son soluciones de la versión homogénea) tiene la forma

$$d_n = A + B\rho^n .$$

Como necesitamos una solución particular para la ecuación completa y lhs es una constante, comenzamos a intentar establecer  $d_n = C$ :

$$pC - C + qC = (p + q)C - C = C - C \neq -1$$

A continuación probamos  $d_n = Cn$ :

$$\begin{aligned} pC(n+1) - Cn + qC(n-1) &= pCn + pC - Cn + qCn - qC \\ &= Cn - Cn + (p-q)C \\ &= (p-q)C = -1 \end{aligned}$$

es decir  $C = \frac{-1}{p-q}$  y la solución particular se convierte en

$$d_n = A + B\rho^n + Cn = A + B\rho^n - \frac{n}{p-q} .$$

Luego aplicamos las **condiciones de contorno**.  $d_0 = 0, d_m = 0$  encontrar  $A$  y  $B$ :

$$\begin{aligned} d_0 &= A + B\rho^0 - \frac{0}{p-q} = A + B = 0 \\ d_m &= A + B\rho^m - \frac{m}{p-q} = A + B\rho^m - \frac{m}{p-q} = 0 , \end{aligned}$$

$$A = -B \quad B + B\rho^m = \frac{m}{p-q} \Rightarrow (\rho^m - 1)B = \frac{m}{p-q} \Rightarrow B = \frac{1}{\rho^m - 1} \cdot \frac{m}{p-q} .$$

Entonces

[Saltar al contenido principal](#)

$$\begin{aligned}
 d_n &= A + B\rho^n - \frac{m}{p-q} \\
 &= -\frac{1}{\rho^m - 1} \cdot \frac{m}{p-q} + \frac{\rho^n}{\rho^m - 1} \cdot \frac{m}{p-q} - \frac{m}{p-q} \\
 &= \frac{1}{p-q} \left( m \frac{\rho^n}{\rho^m - 1} - n \right)
 \end{aligned}$$

- Si  $\rho = 1$ , es decir  $p = q = 1/2$ , la solución general ( $\lambda_1 = 1, \lambda_2 = 1$  son soluciones de la versión homogénea) tiene la forma

$$d_n = A + nB.$$

Para obtener una solución particular encontramos que tanto las conjeturas educadas (**antsazs**)  $d_i = Cyd_i = nC$  no funcionan. Intentamos  $d_i = n^2C$ :

$$\begin{aligned}
 -1 &= pC(n^2 + 1 + 2n) - Cn^2 + qC(n-1)^2 \\
 &= pCn^2 + pC + 2pCn - Cn^2 + qC(n-1)^2 \\
 &= p(Cn^2 + C + 2Cn) - Cn^2 + q(Cn^2 + C - 2Cn) \\
 &= p(Cn^2 + C + 2Cn) - Cn^2 + q(Cn^2 + C - 2Cn) \\
 &= \frac{1}{2}(Cn^2 + C + 2Cn) - Cn^2 + \frac{1}{2}(Cn^2 + C - 2Cn) \\
 &= C.
 \end{aligned}$$

y la solución general resultante es:

$$d_n = A + nB + Cn^2 = A + nB - n^2.$$

Luego, explotamos nuevamente las **condiciones de contorno**.  $d_0 = 0, d_m = 0$  encontrar  $A$  y  $B$ :

$$\begin{aligned}
 d_0 &= A + 0B - n^2 = A = 0 \\
 d_m &= A + mB - n^2 = mB - m^2 = 0 \Rightarrow B = m,
 \end{aligned}$$

Y para  $A = 0, B = m$  la solución general es

$$d_n = A + nB - n^2 = mn - n^2 = n(m-n),$$

lo que nos dice claramente que el tiempo de golpe denes  $O(n^2)$  por  $p = q = 1/2$  ya que tenemos la misma probabilidad de ir y venir.

resumiendo

$$d_n = \begin{cases} \frac{1}{p-q} \left( m \frac{\rho^n}{\rho^m - 1} - n \right) & \text{if } p \neq q \\ n(m-n) & \text{if } p = q \end{cases} \quad i.e. \quad d_n = \begin{cases} \frac{1}{p-q} \left( m \frac{\left(\frac{q}{p}\right)^n - 1}{\left(\frac{q}{p}\right)^m - 1} - n \right) & \text{if } p \neq q \\ n(m-n) & \text{if } p = q \end{cases}$$

**Resultado de la caminata ubiased**. Si  $p = q = 1/2$  el momento del golpe denes  $O(n^2)$  ya que tenemos la misma probabilidad de ir y venir.

El comportamiento para  $m \rightarrow \infty$  es obvio:

$$\lim_{m \rightarrow \infty} n(m-n) = \infty,$$

desde  $m \rightarrow \infty$  vuelve imposible llegar desde  $n$ .

**Resultado del paseo sesgado**. Si  $p \neq q$  y  $p \ll q$ , el camino se desvía hacia la izquierda y esto significa que  $\rho \gg 1$ , mse

[Saltar al contenido principal](#)

golpe den.

En realidad, el comportamiento dem → ∞ es:

$$\begin{aligned}\lim_{m \rightarrow \infty} \frac{1}{p-q} \left( m \frac{\rho^n}{\rho^m - 1} - n \right) &= \lim_{m \rightarrow \infty} \frac{1}{p-q} \left( m \frac{\frac{\rho^n}{\rho^m}}{1 - \frac{1}{\rho^m}} - n \right) \\ &= \lim_{m \rightarrow \infty} \frac{1}{p-q} \left( m \frac{\frac{\rho^n}{\rho^m}}{1 - \frac{1}{\rho^m}} - n \right) \\ &= \lim_{m \rightarrow \infty} \frac{1}{p-q} \left( m \frac{\rho^n}{\rho^m} - n \right)\end{aligned}$$

Tenemos dos casos:

- Si  $q > p$ , entonces  $\rho^m \gg m$  y  $\lim_{m \rightarrow \infty} \frac{1}{p-q} \left( m \frac{\rho^n}{\rho^m} - n \right) = \frac{1}{p-q} (-n) = \frac{1}{q-p} (n)$ . Entonces, el tiempo de golpe es una fracción den.
- Si  $p > q$ , entonces  $\rho^m \ll m$  y está amplificado wrt  $n$ , aumentando el tiempo de golpeo.

**La ruina del jugador**. El paseo del borracho también puede interpretarse en términos del modelo clásico de la ruina del jugador. La idea es la siguiente.

- Tenemos dos jugadores, Alice y Bob, comenzando respectivamente con  $a$  y  $b$  dólares, donde  $a + b = m$ .
- ellos apuestan 1 dólar en cada paso.
- Alice gana con probabilidad  $p$  y Bob gana con probabilidad  $q$ , donde  $p + q = 1$ . Ganar implica aumentar la fortuna personal en 1 dólar de la fortuna del otro, es decir, si Alice gana la primera ronda, el estado de sus respectivas fortunas será  $(a + 1, b - 1) = m$ . Si Bob lo hace, el estado es  $(a - 1, b + 1) = m$ .
- Por lo tanto, alcanzar 0 significa que Alice está arruinada mientras alcanza  $m$ . Significa que Bob está arruinado. En ambos casos paramos el juego.

**Ejercicio**. La ruina del jugador se puede modificar ligeramente para convertirse en una variante de una **cadena de nacimiento y muerte**. Supongamos que tenemos es  $p \geq 0, q \geq 0$  y  $p + q < 1$ . En este problema,  $p$  se llama **tasa de natalidad**, mientras que  $q$  es la **tasa de mortalidad**. Para simular una población estable, en cada estado tenemos la probabilidad de  $r = 1 - p - q$  sin nacimientos ni muertes. si llegamos 0 tenemos **extinción** y si llegamos  $m$  tenemos **supervivencia**. Este se formula de la siguiente manera:

$$x_{n+1} = \begin{cases} x_n + 1 & \text{with probability } p \text{ if } 1 \leq n \leq m - 1 \\ x_n - 1 & \text{with probability } q \text{ if } 1 \leq n \leq m - 1 \\ x_n & \text{with probability } r = 1 - p - q \text{ if } 1 \leq n \leq m - 1 \\ 0 & \text{if } n = 0 \\ m & \text{if } n = m \end{cases}$$

El ejercicio consiste en responder las preguntas Q1 y Q2 de esta cadena. **P1. ¿Probabilidad de extinción?** 1) Tenemos 3 eventos posibles para el teorema de probabilidad total

$$\begin{aligned}p(\text{Extinction}) &= p(\text{1st death})p(\text{Extinction}|\text{1st death}) \\ &\quad + p(\text{1st stable})p(\text{Extinction}|\text{1st stable}) \\ &\quad + p(\text{1st birth})p(\text{Extinction}|\text{1st birth}) \\ &= q \cdot p(\text{Extinction}|\text{1st death}) \\ &\quad + r \cdot p(\text{Extinction}|\text{1st stable}) \\ &\quad + p \cdot p(\text{Extinction}|\text{1st birth}).\end{aligned}$$

[Saltar al contenido principal](#)

$$x_n = q \cdot x_{n-1} + r \cdot x_n + p \cdot x_{n+1} \Rightarrow q \cdot x_{n-1} + (r - 1) \cdot x_n + p \cdot x_{n+1} = 0$$

En realidad, tenemos las condiciones  $x_0 = 1$  y  $x_m = 0$  wrt extinción. Como  $r = 1 - p - q$ , entonces  $r - 1 = 1 - p - q - 1 = -(p + q)$

$q \cdot x_{n-1} - (p + q) \cdot x_n + p \cdot x_{n+1} = 0$  s.t.  $x_0 = 1$ ,  $x_m = 0$ . Aparentemente, la ecuación homogénea anterior es diferente de la de la ruina del jugador (o del paseo del borracho). Sin embargo, si nos fijamos en las soluciones, tenemos dos raíces ( $1$  y  $p$ ) como de costumbre, ya que

$$\begin{aligned} \frac{1}{\lambda^{n-1}} (q\lambda^{n-1} - (p + q)\lambda^n + p\lambda^{n+1}) &= 0 \\ q - (p + q)\lambda + p\lambda^2 &= 0 \\ p\lambda^2 - (p + q)\lambda + q &= 0 \end{aligned}$$

y por Ruffini, tenemos eso  $\lambda_1 = 1$  y como un resultado  $\rho - q = 0 \Rightarrow \lambda_2 = q/p = \rho$ .

Tendremos las siguientes soluciones genéricas (en el primer caso las soluciones son distintas y en el segundo caso tenemos soluciones repetidas):

$$x_n = \begin{cases} A + B\rho^n & \text{if } \rho \neq 1 \\ C + nD & \text{if } \rho = 1. \end{cases} \quad \text{De las dos condiciones de contorno } (x_0 = 1 \text{ y } x_m = 0) \text{ invertido respecto al andar del borracho,}$$

podemos obtener  $A$ ,  $B$ ,  $C$  y  $D$  lo que conduce a probabilidades invertidas en el camino de los borrachos:

$$x_n = \begin{cases} \frac{\rho^n - \rho^m}{1 - \rho^m} & \text{if } p \neq q \\ 1 - \frac{n}{m} & \text{if } p = q \end{cases} \quad i.e. \quad x_n = \begin{cases} \frac{\left(\frac{q}{p}\right)^n - \left(\frac{q}{p}\right)^m}{1 - \left(\frac{q}{p}\right)^m} & \text{if } p \neq q \\ 1 - \frac{n}{m} & \text{if } p = q \end{cases}$$

**Sin embargo**, lo que es diferente aquí con respecto al andar del borracho es que  $p + q < 1$ . Esto significa que pueden ser arbitrariamente pequeños, por ejemplo  $q = 0.1$  y  $p = 0.05$  lo que lleva a  $\rho = 2$ , o  $q = 0.05$  y  $p = 0.1$  lo que lleva a  $\rho = 1/2$ . El resultado es el mismo siempre que  $\rho = 2$  y  $\rho = 1/2$  (por ejemplo  $q = 0.3$ ,  $p = 0.6$  y  $q = 0.6$ ,  $p = 0.3$  respectivamente). En otras palabras, el MCRW es invariante a los cambios del  $r = 1 - p - q$  probabilidades siempre que la proporción  $q/p$  sostiene. **P1. ¿Es hora de sobrevivir?** Como de costumbre, comenzamos formulando la **condición en el primer paso**:

$$\begin{aligned} E(\text{Survival}) &= p(1\text{st death})p(\text{Survival}|1\text{st death}) \\ &\quad + p(1\text{st stable})p(\text{Survival}|1\text{st stable}) \\ &\quad + p(1\text{st birth})p(\text{Survival}|1\text{st birth}) \\ &= q \cdot p(\text{Survival}|1\text{st death}) \\ &\quad + r \cdot p(\text{Survival}|1\text{st stable}) \\ &\quad + p \cdot p(\text{Survival}|1\text{st birth}). \end{aligned}$$

Esto nos lleva a la siguiente **relación de recurrencia no homogénea**:

$$d_n = p(1 + d_{n+1}) + r(1 + d_n) + q(1 + d_{n-1}).$$

$$Dep + q + r = 1 \text{ y } r - 1 = 1 - p - q - 1 = -(p + q) \text{ tenemos}$$

$$pd_{n+1} - (p + q)d_n + qd_{n-1} = -1 \text{ subject to } d_0 = 0, d_m = 0.$$

Las raíces de su **versión homogénea**  $\lambda_1 = 1$  y  $\lambda_2 = \rho = q/p$ .

Si  $\rho \neq 1$ , la solución general tiene la misma forma que la del camino de Druknard (verifíquelo como ejercicio adicional):

[Saltar al contenido principal](#)

$$\begin{aligned}
 d_n &= A + B\rho^n - \frac{m}{p-q} \\
 &= -\frac{1}{\rho^m - 1} \cdot \frac{m}{p-q} + \frac{\rho^n}{\rho^m - 1} \cdot \frac{m}{p-q} - \frac{m}{p-q} \\
 &= \frac{1}{p-q} \left( m \frac{\rho^n}{\rho^m - 1} - n \right)
 \end{aligned}$$

Nuevamente la diferencia con el andar del borracho aparece al definir  $p$  y  $q$ . Tenga en cuenta que si  $p$  y  $q$  son muy pequeños (cercaos a cero), entonces el tiempo de impacto tiende a  $\infty$  incluso cuando las proporciones  $p$  y  $q$  mantienen: mira la fracción  $\frac{1}{p-q}$  dominando el tiempo de golpe para  $p \neq q$ . Hay una buena explicación: la caminata pasa una cantidad importante de tiempo sin avanzar ni retroceder ya que  $1 - p - q \approx 1$ .

Sin embargo, si  $p = q$  entonces  $r = 1 - 2p = 1 - 2q$  de la recurrencia no homogénea anterior

$$pd_{n+1} - (p+q)d_n + qd_{n-1} = -1 \text{ subject to } d_0 = 0, d_m = 0.$$

$$\text{tenemos: } pd_{n+1} - 2pd_n + pd_{n-1} = -1 \text{ subject to } d_0 = 0, d_m = 0.$$

Entonces, la solución para la versión homogénea.  $pd_{n+1} - 2pd_n + pd_{n-1} = 0$

Ahora tiene una doble solución.  $\lambda_1 = \lambda_2 = 1$  y la solución general se convierte en  $d_n = A + nB$ .

Y las conjeturas fundamentadas (antsazs) para  $d_i = Cyd_i = Cn$  no funcionan. Intentemos  $d_i = Cn^2$ :

$$\begin{aligned}
 -1 &= pC(n+1)^2 - 2pCn^2 + pC(n-1)^2 \\
 &= pC(n^2 + 1 + 2n) - 2pCn^2 + pC(n^2 + 1 - 2n) \\
 &= pCn^2 + pC + 2pCn - 2pCn^2 + pCn^2 + pC - 2pCn \\
 &= \cancel{pCn^2} + pC + 2pCn - \cancel{2pCn^2} + \cancel{pCn^2} + pC - 2pCn \\
 &= pC + 2pCn + pC - 2pCn \\
 &= 2pC
 \end{aligned}$$

es decir  $C = \frac{-1}{2p}$  y la **solución general** resultante tiene la siguiente forma:

$$d_n = A + nB + Cn^2 = A + nB - \frac{n^2}{2p}.$$

Y explotamos las **condiciones de contorno**.  $d_0 = 0$  y  $d_m = 0$  encontrar  $A$  y  $B$ :

$$d_0 = A + 0B - C0^2 = 0 \Rightarrow A = 0$$

$$d_m = A + mB - \frac{m^2}{2p} = 0 \Rightarrow B = \frac{m}{2p}$$

Y la **solución general** para  $A = 0$  y  $B = m/2p$  se convierte

$$\begin{aligned}
 d_n &= A + nB - \frac{n^2}{2p} = n \frac{m}{2p} - \frac{n^2}{2p} \\
 &= \frac{1}{2p}(nm - n^2) \\
 &= \frac{1}{2p}n(m - n).
 \end{aligned}$$

En realidad, esto resulta en una normalización por  $2pd(m-n)$  (la solución para el andar del borracho). La interpretación natural es que como  $p \rightarrow 0$  tenemos  $r \rightarrow 1$  y el tiempo de golpeo tiende a  $\infty$ .

## 4.3. Paseos aleatorios en 2D

El enfoque de relaciones de recurrencia es muy útil para **procesos aleatorios 1D** como el paseo del borracho o la ruina del jugador, procesos de nacimiento-muerte y muchos más. Sin embargo, es en el dominio de gráficos más generales como **cuadráculos o polígonos** (por ejemplo, el triángulo de Pascal) donde los paseos aleatorios se vuelven más útiles para los

[Saltar al contenido principal](#)

investigadores y profesionales de la IA. Por ejemplo, una imagen puede verse como una matriz de píxeles donde cada píxel interno  $(i, j)$  (columna, fila) está conectado con 8 vecinos:

$$\begin{array}{ccc} (i-1, j+1) & (i, j+1) & (i+1, j+1) \\ (i-1, j) & (i, j) & (i+1, j) \\ (i-1, j-1) & (i, j-1) & (i+1, j-1) \end{array}$$

- 4 direcciones rectas: West, North, East, South, respectivamente  $(i-1, j)$ ,  $(i, j+1)$ ,  $(i+1, j)$  y  $(i, j-1)$ .
- 4 diagonales:  $(i-1, j+1)$ ,  $(i+1, j+1)$ ,  $(i+1, j-1)$  y  $(i-1, j-1)$ .

Por lo tanto, las imágenes son 8 – neighborhood **cuadrículas** donde nos interesa encontrar objetos, como órganos en imágenes médicas. La tarea se llama **segmentación** y para imágenes médicas, que suelen ser muy ruidosas, es muy útil hacer clic en algunos píxeles del órgano y en algunos píxeles fuera de él para **etiquetar** los píxeles que pertenecen al órgano. Nuestro colega Leo Grady desarrolló una tecnología para segmentar imágenes médicas en su artículo [Random Walks for Image Segmentation](#) y la transfirió a Siemens.

**La sala de escape**. Resolvamos un problema más simple (pero conceptual y metodológicamente idéntico) que involucra un 4 – neighborhood **cuadrícula** (sólo direcciones rectas).

En [la Fig. 4.5](#) creamos un pequeño juego 2D. Tenemos una 4–cuadrícula cuyos nodos (llamados **nodos interiores**) son las posiciones  $x_1, x_2, \dots, x_5$  de un jugador en una habitación. El gráfico también muestra en un color más vivo las puertas. Algunas de las puertas son *Salidas*.  $E_1, E_2, \dots, E_6$  y algunas otras puertas están controladas por *policías*.  $P_1, P_2, P_3$ .

El juego es el siguiente:

¿Cuál es la probabilidad de escapar de cada posición?

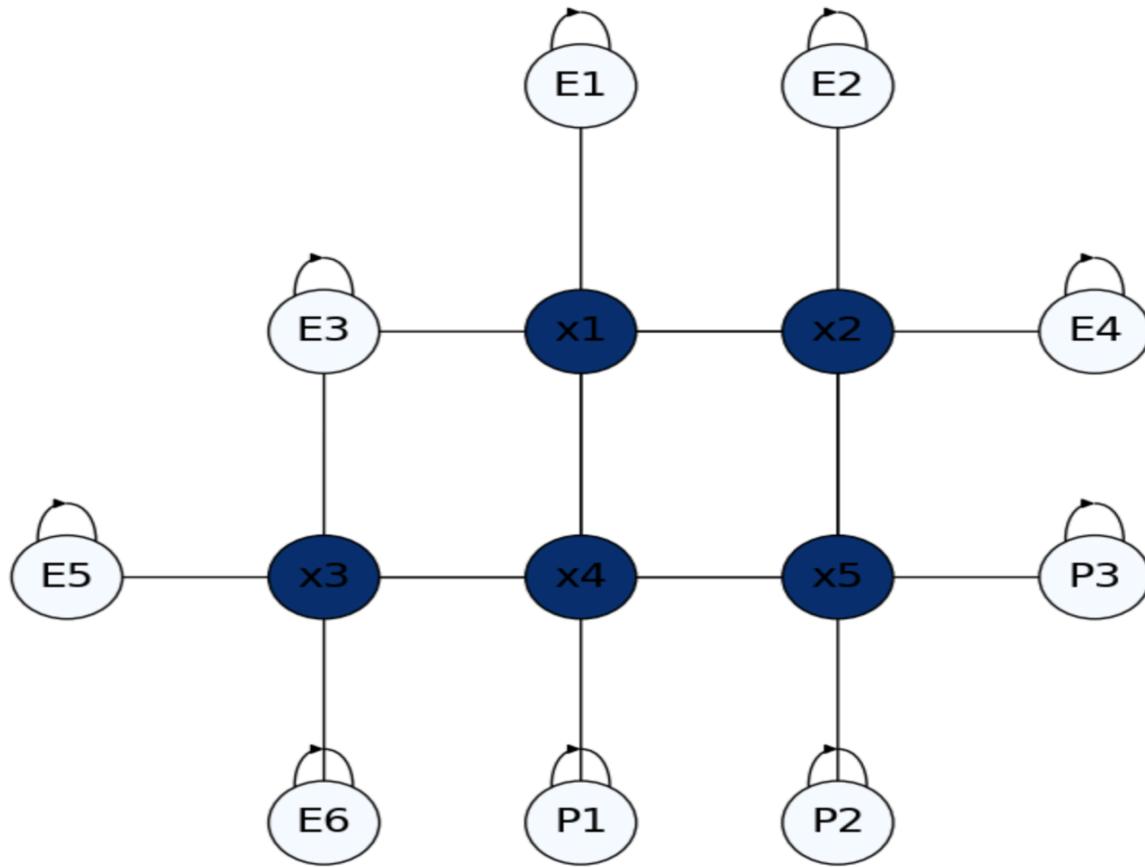


Fig. 4.5 Sala de escape con salidas  $E_i$  y policías  $P_i$  como nodos fronterizos.

Dado que las salidas y los policías son **estados absorbentes**, esta es una versión 2D del paseo del borracho. El **juego termina** cuando llegamos a una Salida, con recompensa 1, o a un policía, con recompensa 0. Por lo tanto, nuestro problema es estimar  $p(\text{Exit}|x_i)$  para todos  $x_i$ .

[Saltar al contenido principal](#)

Imagine que el jugador comienza en el nodo  $x_4$  (cerca de un policía y lejos de una salida). Es razonable tener  $p(\text{Exit}|x_4) < p(\text{Exit}|x_1)$ . En realidad, si el jugador "ve" al policía, huirá, pero ¿adónde? El jugador no conoce de antemano la posición de las salidas ni de los demás policías. Esta información le será **transmitida** desde los llamados **nodos fronterizos**: los nodos policiales "enviarán" una recompensa de 0 mientras que las salidas enviarán una recompensa de 1, a través de sus respectivas condiciones de contorno.

**Principio Armónico** . Dejar  $f(i) = p(\text{Exit}|i)$  donde  $j$  puede ser un  $x_j$  o  $E_j$  entonces, asumimos que la probabilidad de un nodo interno es el promedio de la de sus 4 vecinos:

$$f(i) = \frac{1}{4} \sum_{j \in \mathcal{N}_i} f(j) .$$

En realidad, tenemos un **sistema lineal** con 5 incógnitas  $f(x_1), \dots, f(x_5)$  y 5 ecuaciones:

$$\begin{aligned} f(x_1) &= \frac{1}{4}[f(E_3) + f(E_1) + f(x_2) + f(x_4)] = \frac{1}{4}[1 + 1 + f(x_2) + f(x_4)] \\ f(x_2) &= \frac{1}{4}[f(x_1) + f(E_2) + f(E_4) + f(x_5)] = \frac{1}{4}[f(x_1) + 1 + 1 + f(x_5)] \\ f(x_3) &= \frac{1}{4}[f(E_5) + f(E_3) + f(x_4) + f(E_6)] = \frac{1}{4}[1 + 1 + f(x_4) + 1] \\ f(x_4) &= \frac{1}{4}[f(x_3) + f(x_1) + f(x_5) + f(P_1)] = \frac{1}{4}[f(x_3) + f(x_1) + f(x_5) + 0] \\ f(x_5) &= \frac{1}{4}[f(x_4) + f(x_2) + f(P_3) + f(P_2)] = \frac{1}{4}[f(x_4) + f(x_2) + 0 + 0] \end{aligned}$$

La "armonidad" asegura que el sistema tiene una **solución única** y podemos resolverla muy fácilmente si entendemos la estructura del sistema. A continuación, incluso cuando usamos una inversa, observamos que **en realidad no necesitamos calcularla manualmente en los ejercicios, sino aproximar la solución de forma iterativa**.

**Matriz de Transición** . Dejar  $P$  ser una  $n \times n$  matriz, donde  $n = |V|$  es el número de nodos del gráfico  $G = (V, E)$ . Entonces, el componente  $p_{ij}$  es la probabilidad de llegar al nodo  $j$  desde el nodo  $i$ . Para la **sala de escape** anterior, la matriz de transición tiene la siguiente **estructura de bloques**:

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_{n_B} & \mathbf{0}_{n_B \times n_I} \\ \mathbf{R}_{n_I \times n_B} & \mathbf{Q}_{n_I} \end{bmatrix}$$

dónden  $= n_B + n_I$  y  $n_B$  es el número de nodos **fronterizos** (absorbentes) y  $n_I$  es el número de nodos **interiores**. Entonces nosotros tenemos:

- $\mathbf{I}_{n_B}$  es la **matriz identidad de la dimensión**  $n_B$ . Tiene las probabilidades entre nodos fronterizos: solo bucles automáticos,  $p_{ii} = 1$  en diagonal y  $p_{ij} = 0$  para  $j \neq i$ .
- $\mathbf{R}_{n_I \times n_B}$  tiene las probabilidades entre nodos interiores ( $x_1, \dots, x_5$ ) **en las filas** y nodos de borde ( $E_1, \dots, E_6, P_1, P_2, P_3$ ) **en las columnas**. En nuestro ejemplo, tenemos  $p_{ij} = 1/4$  cuando  $j$  es vecino de  $i$ :

$$\mathbf{R} = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

- $\mathbf{Q}_{n_I}$  es una **matriz cuadrada** con probabilidades solo entre nodos interiores ( $x_1, \dots, x_5$ ). Si alguna.

$$\mathbf{Q} = \begin{bmatrix} 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \end{bmatrix}$$

Una **propiedad** interesante de  $\mathbf{P}$  es que *todas las filas deben sumar 1* (*estocástico simple*). Luego, observe que la suma de probabilidades de la misma fila en  $\mathbf{R}$  y  $\mathbf{Q}$  es en realidad 1 por cada fila.

¿Cuál es la relación entre las dos matrices anteriores?

La estocasticidad simple de  $\mathbf{P}$  da una idea precisa de cómo vincular  $\mathbf{R}$  y  $\mathbf{Q}$  a través de un sistema lineal.

- Por un lado tenemos eso  $\mathbf{I}_{n_I} - \mathbf{Q}$  (matriz de identidad de tamaño  $n_I$  menos  $\mathbf{R}$ ) codifica, sumando cada fila, las probabilidades de "escapar hacia cualquier nodo de absorción". Por ejemplo, la probabilidad de "escapar" de  $x_2$  hacia cualquiera de los nodos de absorción es  $-\frac{1}{4} + 1 - \frac{1}{4} = \frac{2}{4} = \frac{1}{2}$ :

$$\mathbf{I}_{n_I} - \mathbf{Q} = \begin{bmatrix} 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 1 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 0 & -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 \end{bmatrix}$$

- Dejar  $f = [f_B \ f_D]^T$  sean las probabilidades condicionales  $f(i) = p(\text{Exit}|i)$  donde algunos de ellos son conocidos ( $f_B = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0]$ ) y algunos otros deben estimarse ( $f_D$ ). Con un pequeño abuso de notación, denotemos  $f(x_i)$  como  $x_i$ . Entonces nosotros tenemos  $f_D = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]$ .
- Luego, desde el otro lado, el producto matricial  $\mathbf{R}f_B$  codifica las probabilidades de pasar de cada nodo interior a cualquier borde:

$$\mathbf{R}f_B = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} + \frac{1}{4} \\ \frac{1}{4} + \frac{1}{4} \\ \frac{1}{4} + \frac{1}{4} + \frac{1}{4} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{3}{4} \\ 0 \\ 0 \end{bmatrix}$$

Como resultado tenemos el sistema lineal:

$$\underbrace{(\mathbf{I}_{n_I} - \mathbf{Q})}_{\mathbf{A}} \underbrace{f_D}_{\mathbf{x}} = \underbrace{\mathbf{R}f_B}_{\mathbf{b}}.$$

El sistema correspondiente en el ejemplo es:

$$\begin{bmatrix} 1 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & 0 & 0 & -\frac{1}{4} \\ 0 & 0 & 1 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 0 & -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{3}{4} \\ 0 \\ 0 \end{bmatrix}$$

**Solución iterativa** . Este sistema, que es equivalente al dado por la restricción armónica, es muy adecuado para una solución

[Saltar al contenido principal](#)

En este sentido se cumplen los requisitos de convergencia del conocido **Algoritmo de Jacobi**: básicamente el **A** es *diagonalmente dominante* (sus entradas son mayores en valor absoluto que las fuera de la diagonal). Este algoritmo obedece a la siguiente relación de recurrencia:

$$\mathbf{x}^{t+1} = \mathbf{D}^{-1}\mathbf{b} - \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^t ,$$

dónde:

- $\mathbf{D}^{-1}$  es la inversa de  $diag(\mathbf{A})$ . En nuestro caso es  $\mathbf{I}_{n_I}$ .
- $\mathbf{U}$  y  $\mathbf{L}$  son, respectivamente, las submatrices triangular superior e inferior de  $\mathbf{A}$  (diagonal no incluida, es decir, puesta a cero).

Básicamente, en este problema, el algoritmo de Jacobi es extremadamente simple:

$$\mathbf{x}^{t+1} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^t .$$

Así, estableciendo  $\mathbf{x}_0 = [1 \ 1 \ 1 \ 1 \ 1]^T$  tenemos:

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \\ .75 \\ .5 \end{bmatrix} \Rightarrow \mathbf{x}_2 = \begin{bmatrix} .9375 \\ .875 \\ .625 \\ .4375 \end{bmatrix} \Rightarrow \mathbf{x}_3 = \begin{bmatrix} .875 \\ .84375 \\ .578125 \\ .375 \end{bmatrix} \Rightarrow \mathbf{x}_4 = \begin{bmatrix} .85546875 \\ .8125 \\ .5390625 \\ .35546875 \end{bmatrix} ,$$

que es una buena aproximación de la **solución exacta**:

$$\mathbf{x} = \begin{bmatrix} .82303371 \\ .78651685 \\ .87640449 \\ .50561798 \\ .32303371 \end{bmatrix} .$$

Como podemos ver en [la Fig. 4.6](#), los nodos más cercanos a los Policias tienen menos probabilidad de éxito. Quizás el ejemplo más ilustrativo sea  $x_4 \approx 0.5$ ; A medio camino entre escapar y capturar!

[Saltar al contenido principal](#)

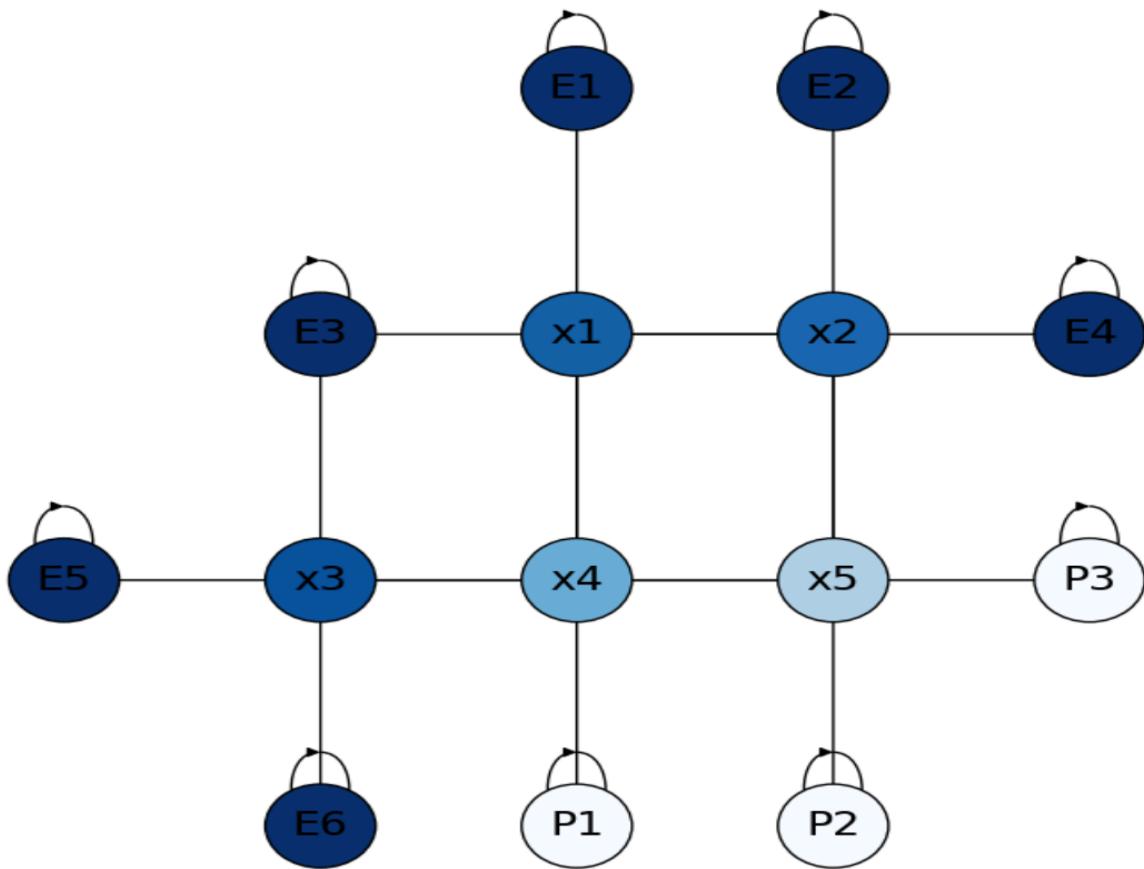


Fig. 4.6 Solución al Escape Room.

Obviamente,  $six_i = p(\text{Exit}|i)$ , entonces  $1 - x_i = p(\text{Policemen}|i)$ .

**Ventajas de calcular la inversa** . lo inverso  $(\mathbf{I}_{n_I} - \mathbf{Q})^{-1}$  se llama **matriz fundamental N** de la cadena de Markov (MC). las entradas  $N_{ij}$  puede interpretarse como el **número esperado de veces que el MC estará en estado jantes de la absorción cuando está en estado i**.

En nuestro ejemplo, tenemos:

$$(\mathbf{I}_{n_I} - \mathbf{Q})^{-1} = \begin{bmatrix} 1.1741573 & 0.33707865 & 0.08988764 & 0.35955056 & 0.1741573 \\ 0.33707865 & 1.16853933 & 0.04494382 & 0.17977528 & 0.33707865 \\ 0.08988764 & 0.04494382 & 1.07865169 & 0.31460674 & 0.08988764 \\ 0.35955056 & 0.17977528 & 0.31460674 & 1.25842697 & 0.35955056 \\ 0.1741573 & 0.33707865 & 0.08988764 & 0.35955056 & 1.1741573 \end{bmatrix}$$

En realidad, el producto  $\mathbf{t} = \mathbf{N}\mathbf{1}$  (vector de columna de todos los unos) proporciona el **número esperado de pasos antes de la absorción para cada estado inicial** . A saber

$$\mathbf{t} = \begin{bmatrix} 2.13483146 \\ 2.06741573 \\ 1.61797753 \\ 2.47191011 \\ 2.13483146 \end{bmatrix}.$$

**Solución Montecarlo** . ¿Y si en lugar de resolver un sistema lineal lanzamos **paseos aleatorios** (RW) desde los estados interiores?  $x_1, \dots, x_5$ ? Procedemos de la siguiente manera:

- Para cada  $x_i$  lanzamiento  $n$  paseos de longitud  $l$  de cada  $x_i$ .
- La proporción de caminatas iniciadas desde  $x_i$  alcanzar un estado de salida es una aproximación de  $p(i) = p(\text{Exit}|i)$ .

[Saltar al contenido principal](#)

En términos binomiales,  $p(i)$  puede verse como la “probabilidad de éxito” y sabíamos que cuando  $n \rightarrow \infty$  tenemos una distribución normal de la media  $\mu = np$  y desviación estándar  $\sigma = \sqrt{npq}$ .

Imagine ahora que quiere asegurarse de que la probabilidad de éxito sea .95.

$$p\left(-a \leq \frac{p(i) - np}{\sqrt{npq}} \leq a\right) = p(-a \leq Z \leq a) = .95$$

es decir  $Z$  es una variable estandarizada. Debido a la simetría de la distribución normal, la ecuación anterior significa que las probabilidades de las colas satisfacen  $p(-a \leq Z) = p(Z \geq a) = (1 - 0.95)/2 = 0.025$ . Consultando la [tabla de probabilidad acumulativa normal estándar](#) tenemos que  $a = 1.9 \approx 2$ . Entonces:

$$-2 \leq \frac{p(i) - np}{\sqrt{npq}} \leq 2$$

Normalizando tanto el numerador como el denominador mediante  $n$  tenemos

$$-2 \leq \frac{\frac{p(i)-np}{n}}{\frac{\sqrt{pq}}{\sqrt{n}}} \leq 2$$

llevando a

$$-2 \leq \frac{\frac{p(i)}{n} - p}{\frac{\sqrt{pq}}{\sqrt{n}}} \leq 2 \Rightarrow -2 \leq \frac{\frac{p(i)}{n} - p}{\sqrt{\frac{pq}{n}}} \leq 2 \Rightarrow -2\sqrt{\frac{pq}{n}} \leq \frac{p(i)}{n} - p \leq 2\sqrt{\frac{pq}{n}}.$$

Desde  $\sqrt{pq} < 1/2$  tenemos

$$p\left(-\frac{1}{\sqrt{n}} < \frac{p(i)}{n} - p < \frac{1}{\sqrt{n}}\right) = 0.95.$$

Esto significa que si queremos asegurar que la desviación entre la media  $p(i)$  y la probabilidad de éxito satisface

$$\frac{p(i)}{n} - p < \frac{1}{\sqrt{n}} = 0.01$$

con probabilidad 0.95, nosotros necesitamos  $n = 10,000$  ¡camina por un límite superior tan pequeño! En realidad la solución Monte Carlo obtenida al lanzar 10,000 paseos desde cada nodo interior es

$$\mathbf{x} = \begin{bmatrix} .82758621 \\ .80224404 \\ .84444444 \\ .49731183 \\ .302391 \end{bmatrix}$$

donde la duración de cada caminata es bastante flexible ( $l = 20$  en este caso), ya que los estados de absorción son bastante cercanos.

Las cadenas de Monte Carlo Markov (MCMC) son muy ineficientes aquí, pero a veces se convierten en una estrategia de búsqueda eficaz cuando se basan en datos. [Ejercicio](#) . ¿Qué pasa si tenemos recompensas negativas? En el pequeño problema de [la figura 4.7](#) tenemos [estados absorbentes negativos](#).

$n_1, \dots, n_5$  etiquetado como -1 y [estados absorbentes positivos](#)  $p_1$  y  $p_2$  ajustado a +1. En este caso, tenemos sólo tres estados interiores  $x_1, x_2$  y  $x_3$ . ¿Cuáles son sus probabilidades?

[Saltar al contenido principal](#)

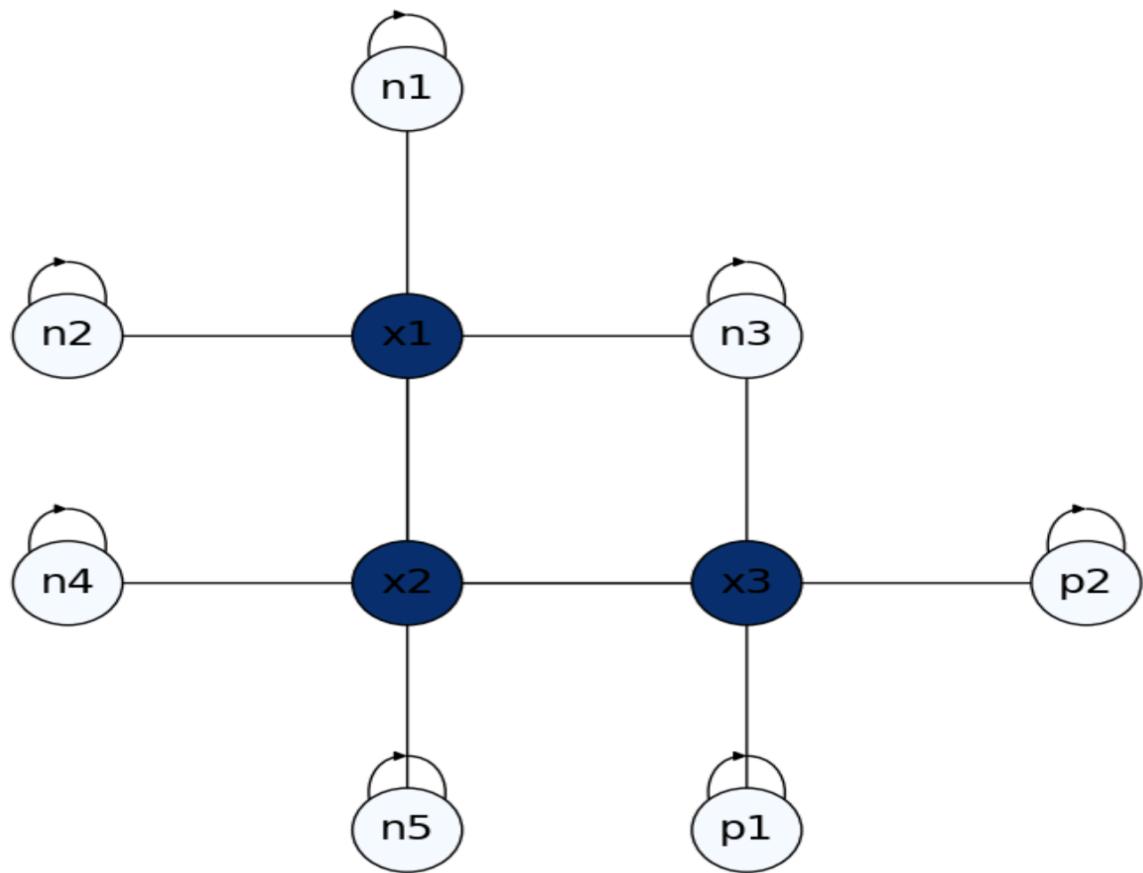


Fig. 4.7 Pequeño Escape Room con recompensas positivas y negativas.

Formulemos las matrices  $\mathbf{Q}$  y  $\mathbf{R}$ . Dado  $n_B = 5 + 2 = 7$  y  $n_I = 3$ , tenemos

$$\mathbf{Q}_3 = \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$

$$\mathbf{R}_{3 \times 7} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

Entonces, desde  $f_B = [-1 \ -1 \ -1 \ -1 \ -1 \ +1 \ +1]^T$  y  $f_D = [x_1 \ x_2 \ x_3]^T$  configuramos el sistema

$$\underbrace{(\mathbf{I}_3 - \mathbf{Q})}_{\mathbf{A}} \underbrace{f_D}_{\mathbf{x}} = \underbrace{\mathbf{R} f_B}_{\mathbf{b}} . \text{ como sigue}$$

$$\begin{bmatrix} 1 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -1 \\ +1 \\ +1 \\ +1 \\ +1 \end{bmatrix}$$

es decir

[Saltar al contenido principal](#)

$$\begin{bmatrix} 1 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -\frac{3}{4} \\ -\frac{2}{4} \\ +\frac{1}{4} \end{bmatrix}$$

Ahora configuremos y resolvamos el sistema **iterativo** vía **Jacobi** para estos problemas:

$$\mathbf{x}^{t+1} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^t.$$

$$\begin{bmatrix} \mathbf{x}_1^{t+1} \\ \mathbf{x}_2^{t+1} \\ \mathbf{x}_3^{t+1} \end{bmatrix} = \begin{bmatrix} -\frac{3}{4} \\ -\frac{2}{4} \\ +\frac{1}{4} \end{bmatrix} - \begin{bmatrix} 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 0 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^t \\ \mathbf{x}_2^t \\ \mathbf{x}_3^t \end{bmatrix}$$

Así, estableciendo  $\mathbf{x}_0 = [1 \ 1 \ 1]^T$  tenemos:

$$\mathbf{x}_1 = \begin{bmatrix} -.5 \\ 0 \\ .5 \end{bmatrix} \Rightarrow \mathbf{x}_2 = \begin{bmatrix} -.75 \\ -.5 \\ .25 \end{bmatrix} \Rightarrow \mathbf{x}_3 = \begin{bmatrix} -.875 \\ -.625 \\ .125 \end{bmatrix} \Rightarrow \mathbf{x}_4 = \begin{bmatrix} -.90625 \\ -.6875 \\ .09375 \end{bmatrix},$$

que es una buena aproximación de la **solución exacta**:

$$\mathbf{x} = \begin{bmatrix} -.92857 \\ -.71428 \\ .071428 \end{bmatrix}.$$

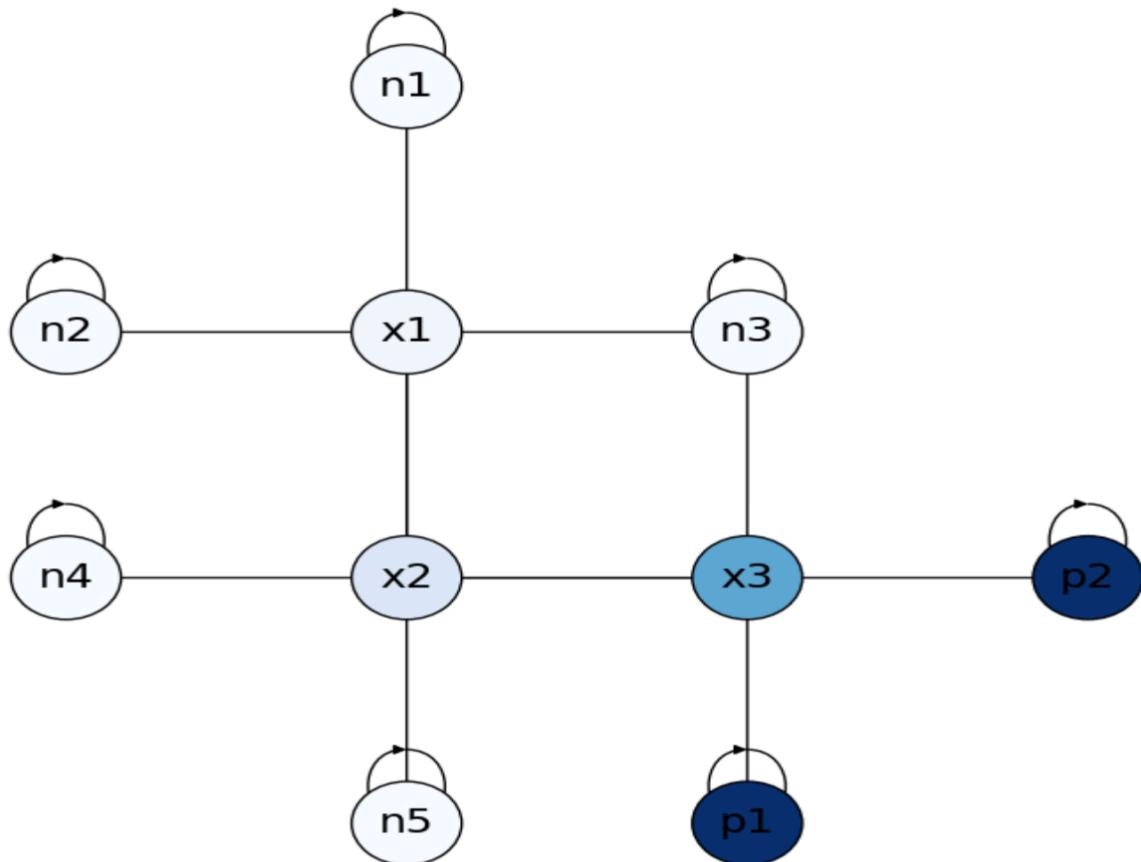


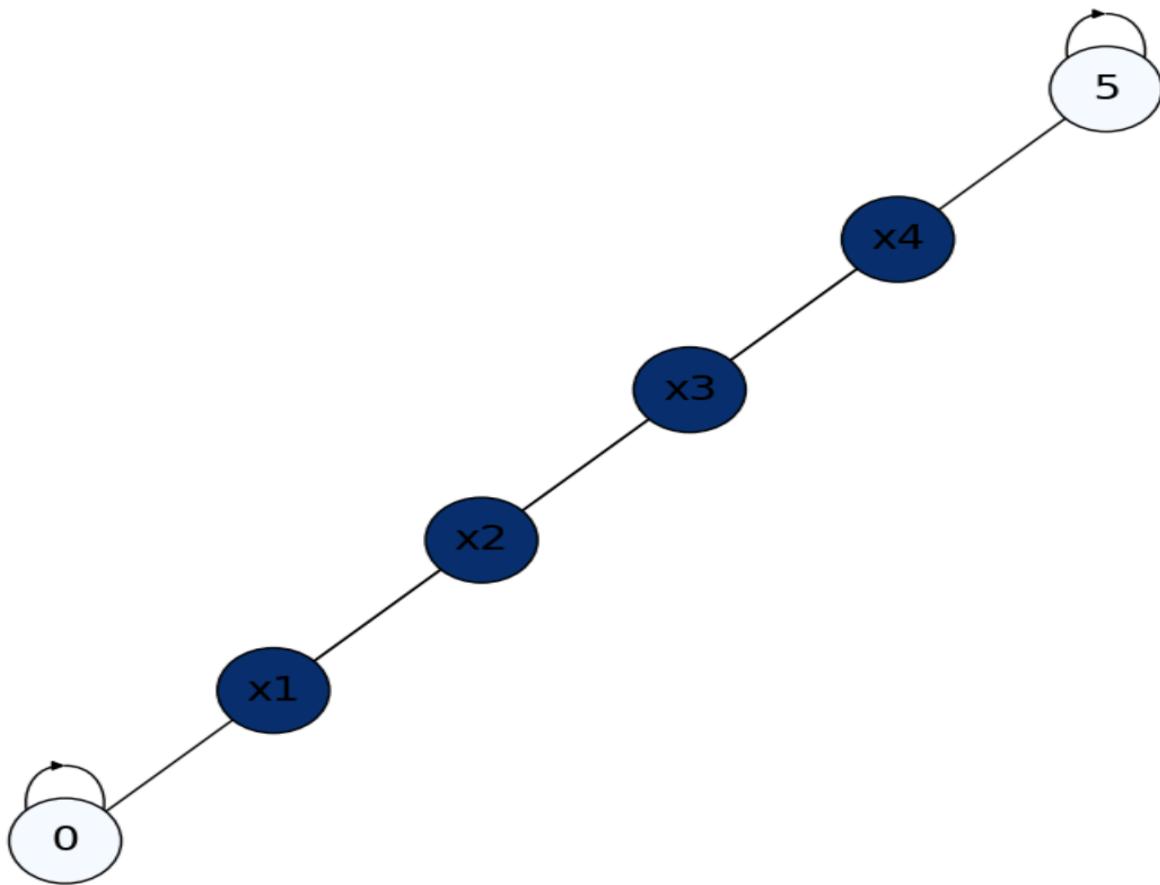
Fig. 4.8 Solución de Escape Room pequeño con recompensas positivas y negativas.

Como podemos ver en la figura 4.8, los estados negativos están más cerca de los estados absorbentes negativos, donde se

[Saltar al contenido principal](#)

Resolvamos el andar del borracho desde esta perspectiva. En [la figura 4.9](#) tenemos **dos estados absorbentes**.

0(Bar) y 1(Home) y cuatro estados del interior  $x_1, x_2, x_3$  y  $x_4$ . Lo sabemos  $p(\text{Home}|x_i) = \frac{x_i}{m}$  con  $m = 5$  en este caso. ¿Cómo demostrarlo utilizando el enfoque 2D?



*Fig. 4.9 Formulación del paseo del borracho con dos estados absorbentes: Barra (0) y Inicio (5).*

Aplicamos la lógica del proceso Markoviano de la siguiente manera.

Formulemos las matrices  $\mathbf{Q}$  (probabilidades entre nodos interiores) y  $\mathbf{R}$  (probabilidades entre nodos interiores y absorbentes). Dado  $n_B = 2$  y  $n_I = 4$ , tenemos

$$\mathbf{Q}_4 = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

$$\mathbf{R}_{4 \times 2} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$$

Entonces, desde  $f_B = [0 \ 1]^T$  y  $f_D = [x_1 \ x_2 \ x_3 \ x_4]^T$  configuramos el sistema

$$\underbrace{(\mathbf{I}_4 - \mathbf{Q})}_{\mathbf{A}} \underbrace{f_D}_{\mathbf{x}} = \underbrace{\mathbf{R} f_B}_{\mathbf{b}} . \text{ como sigue}$$

[Saltar al contenido principal](#)

$$\begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

es decir

$$\begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{2} \end{bmatrix}$$

Ahora configuremos y resolvamos el sistema **iterativo** vía **Jacobi** para este problema:

$$\mathbf{x}^{t+1} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^t .$$

$$\begin{bmatrix} \mathbf{x}_1^{t+1} \\ \mathbf{x}_2^{t+1} \\ \mathbf{x}_3^{t+1} \\ \mathbf{x}_4^{t+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{2} \end{bmatrix} - \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^t \\ \mathbf{x}_2^t \\ \mathbf{x}_3^t \\ \mathbf{x}_4^t \end{bmatrix}$$

Así, estableciendo  $\mathbf{x}_0 = [1 \ 1 \ 1 \ 1]^T$  tenemos:

$$\mathbf{x}_1 = \begin{bmatrix} .5 \\ 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \mathbf{x}_2 = \begin{bmatrix} .5 \\ .75 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \mathbf{x}_3 = \begin{bmatrix} .375 \\ .75 \\ .875 \\ 1 \end{bmatrix} \Rightarrow \mathbf{x}_4 = \begin{bmatrix} .375 \\ .625 \\ .875 \\ .9375 \end{bmatrix},$$

que es una aproximación razonable de la **solución exacta**:

$$\mathbf{x} = \begin{bmatrix} .2 \\ .4 \\ .6 \\ .8 \end{bmatrix}.$$

Consulte [la figura 4.10](#), los estados más oscuros están más cerca de Home. Recuerda que  $1 - x_i$  da las probabilidades frente a la Bar!

[Saltar al contenido principal](#)

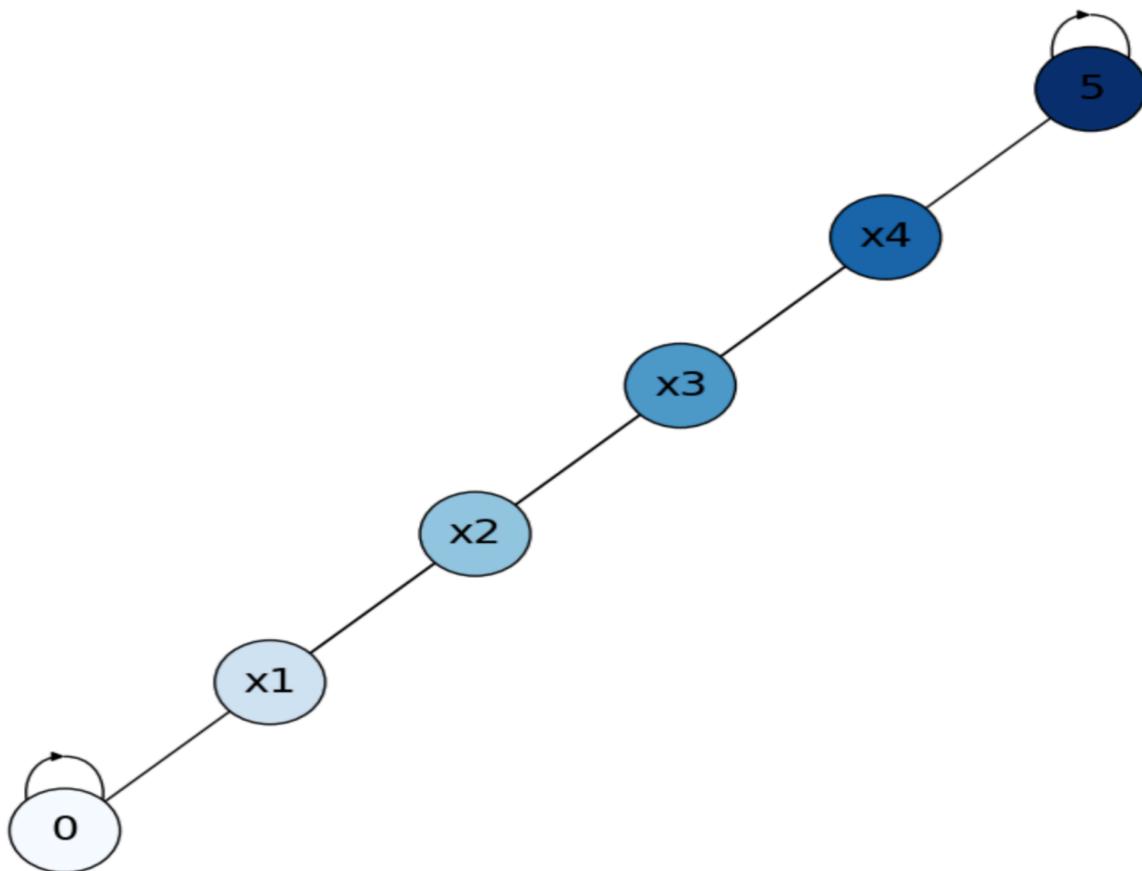


Fig. 4.10 Solución de Escape Room pequeño con recompensas positivas y negativas.

## 4.4. El fenómeno del

### 4.4.1. Cadenas de Markov y equilibrio

Al estudiar las Cadenas de Markov (MC) hemos dejado de lado intencionalmente un aspecto fundamental de ellas: sus **comportamientos a largo plazo**. Esto incluye, por supuesto, sus distribuciones limitantes o estados estacionarios. La búsqueda de la armonía, por ejemplo, nos proporciona cierta búsqueda del equilibrio en el sistema lineal. Véase, por ejemplo, la Fig. 4.8 , donde la solución numérica de este sistema garantiza que el estado de un nodo interior converja al promedio de sus vecinos (que pueden incluir otros nodos interiores o de borde/absorbentes). En realidad, la armonidad implica equilibrio y viceversa, como mostraremos más adelante, como un primer ejemplo de Teoría de Grafos Espectrales.

### 4.4.2. Barajar cartas

Por el momento, una idea aproximada de este concepto es identificar **equilibrio con completo desorden o aleatoriedad** . Consideremos, por ejemplo, una mini baraja de cartas que consta únicamente del  $n = 13$  tarjetas de la traje. Inicialmente, esta baraja se ordena según sus valores nominales crecientes, es decir, tenemos

$$A\heartsuit 2\heartsuit 3\heartsuit 4\heartsuit 5\heartsuit 6\heartsuit 7\heartsuit 8\heartsuit 9\heartsuit 10\heartsuit J\heartsuit Q\heartsuit K\heartsuit$$

Un **riffle shuffling** consta de:

1- *Selecciona un punto de corte*  $k$  (aproxadamente en  $1/2$  del mini-deck) de modo que dividamos el deck en 2 paquetes de tamaño similar:  $1, 2, \dots, k$ ,  $k+1, k+2, \dots, n$ . Para  $k=6$  tenemos:

$$A\heartsuit 2\heartsuit 3\heartsuit 4\heartsuit 5\heartsuit 6\heartsuit \quad 7\heartsuit 8\heartsuit 9\heartsuit 10\heartsuit J\heartsuit Q\heartsuit K\heartsuit$$

2- *Intercala los dos subplatos* para que se conserven las posiciones relativas dentro de cada subplato y luego forme un nuevo miniplato:

$$A\heartsuit 7\heartsuit 2\heartsuit 8\heartsuit 9\heartsuit 3\heartsuit 10\heartsuit 4\heartsuit 5\heartsuit J\heartsuit 6\heartsuit Q\heartsuit K\heartsuit$$

[Saltar al contenido principal](#)

Una **secuencia ascendente** es un subconjunto máximo de tarjetas donde sus valores sucesivos se muestran en orden creciente. Por ejemplo, en el entrelazado anterior tenemos dos secuencias ascendentes:  $A, 2, 3, 4, 5, 6$  y  $7, 8, 9, 10, J, Q, K$ . En realidad, estas secuencias ascendentes provienen del primer corte, por lo que pueden usarse para reconstruir el orden original de la baraja.

¿Qué pasa si realizamos **otro riffle shuffle**? Ahora seleccionamos  $k = 7$  para cortar y tenemos:

$$A\heartsuit \ 7\heartsuit \ 2\heartsuit \ 8\heartsuit \ 9\heartsuit \ 3\heartsuit \ 10\heartsuit \quad 4\heartsuit \ 5\heartsuit \ J\heartsuit \ 6\heartsuit \ Q\heartsuit \ K\heartsuit$$

Antes de intercalar las cartas, tenemos que cada baraja tiene **2 secuencias ascendentes**:

- $A, 2, 3$  y  $7, 8, 9, 10$  para el paquete izquierdo.
- $4, 5, 6$  y  $J, Q, K$  para el paquete correcto.

Esto quiere decir que antes de mezclar las cartas hemos **duplicado** el número de secuencias ascendentes.

Intercalemos los dos paquetes anteriores:

$$A\heartsuit \ 4\heartsuit \ 7\heartsuit \ 5\heartsuit \ 2\heartsuit \ J\heartsuit \ 8\heartsuit \ 6\heartsuit \ 9\heartsuit \ Q\heartsuit \ 3\heartsuit \ K\heartsuit \ 10\heartsuit$$

con secuencias ascendentes:

- $A, 2, 3$ .
- $4, 5, 6$ .
- $7, 8, 9, 10$ .
- $J, Q, K$ .

Efectivamente, cada nuevo **barajado** tiende a duplicar el número de secuencias ascendentes hasta que se alcanza la capacidad del mazo. ¿Qué es tal capacidad? Desde  $2^3 < 13 < 2^4$ , después 4 baraja tendríamos el minimazo **completamente mezclado**.

¿Por qué? Porque para  $a = 4$  barajas rápidas que tenemos 13 secuencias ascendentes, es decir, el mini-deck en orden descendente:

$$K\heartsuit \ Q\heartsuit \ J\heartsuit \ 10\heartsuit \ 9\heartsuit \ 8\heartsuit \ 7\heartsuit \ 6\heartsuit \ 5\heartsuit \ 4\heartsuit \ 3\heartsuit \ 2\heartsuit \ 1\heartsuit$$

si tenemos  $n$  tarjetas, el número máximo de **secuencias ascendentes** es  $n$ , es decir, cada tarjeta  $c_i$  tiene un valor nominal superior a  $c_{i+1}$  para  $i = 1, 2, \dots, n - 1$ . Así para  $n = 52$  tarjetas que necesitamos  $2^a \approx 52$  mezclas rápidas, es decir  $a = 6$ .

Aldous y Diaconis demostraron que necesitamos  $a = 7$  baraja con rapidez para asegurar un desorden total. En cualquier caso tenemos eso a es  $O(\log_2 n)$ .

El **fenómeno de corte** hace referencia a que al medir qué tan cerca estamos de la aleatoriedad total, la mayoría de las veces estamos muy lejos. Sin embargo, en cierto punto, nos acercamos drásticamente a la aleatoriedad. No todas las cadenas de Markov tienen esta propiedad, pero cuando la tienen, ese punto se llama punto **de corte**.

< Anterior  
[3. Probabilidad](#)

Próximo >  
[5. Caminos, flujos y ciclos](#)

# Caminos, flujos y ciclos

## Contenido

- 5.1. Caminos más cortos
- 5.2. Flujo máximo
- 5.3. Ciclos: hamiltoniano y euleriano

El estudio del flujo de información es de suma importancia en la IA. Por ejemplo, en las redes neuronales basadas en gráficos (GNN) modernas, cada **nodo** representa una neurona con un estado que se puede aprender. Los nodos comienzan a tener un **estado aleatorio** que se modifica mediante la recopilación **de mensajes** de sus respectivos vecinos. Cada mensaje se refiere al estado actual del vecino correspondiente. Así, el objetivo del aprendizaje es que los nodos marcados (etiquetados) como pertenecientes a una clase determinada pasen a tener un estado interno similar. Esta tarea se conoce como **clasificación de nodos** y solo se conocen de antemano algunas de las etiquetas de algunos nodos.

En la práctica, los nodos cuyas etiquetas se conocen se consideran nodos absorbentes como en el tema anterior y debemos encontrar las etiquetas en un proceso similar a resolver un sistema lineal. Sin embargo, las GNN **no son lineales** ya que pretenden expresar interacciones más complejas como ocurre en el cerebro humano. Aparte de este comportamiento no lineal, estas redes se basan en gráficos como lo hacen los solucionadores de nodos no absorbentes. De hecho, si cambiamos el gráfico (los enlaces), podemos terminar fallando en la tarea de aprender el mismo estado interno para nodos de la misma clase y, en consecuencia, etiquetar los nodos incorrectamente. En otras palabras, la forma en que se comunican los nodos afecta el desempeño de la tarea de aprendizaje.

¿Cuál es la [mejor forma de comunicar](#) dos nodos? ¿Cuál es la [posición relativa de dos nodos](#) en la red? ¿Algunas redes son más fáciles de navegar que otras? ¿Necesitamos [estructuras auxiliares como árboles](#) para responder a todas estas preguntas?

El [ingeniero de IA \(AIer\)](#) debe dominar técnicas basadas en gráficos para programar, comprender y explotar flujos de información.

## 5.1. Caminos más

Generalmente es interesante saber [a qué distancia](#) se encuentra un nodo determinado.*i*de otro en el mismo gráfico. Consideremos, por ejemplo, el caso sencillo de una cuadrícula 2D. Si la cuadrícula es una 4—vecindario (4N) uno, todos los caminos tienen la misma longitud, es decir  $m + n$ , donde  $m$  es el número de movimientos horizontales y  $n$  es el número de movimientos verticales.

Considere, por ejemplo, el problema de encontrar el **camino más corto** (SP) entre nodos  $i = (a, b)$  y  $j = (m, n)$  en un **gráfico más general** incluso donde  $i$  y  $j$  no tienen asociado ningún atributo posicional . En este caso, **más corto** significa **un número mínimo de aristas entre los dos nodos** . En otras palabras, queremos descubrir cómo [navegar de manera óptima entre dos nodos](#) utilizando exclusivamente los bordes del gráfico .

Por ejemplo, si permitiéramos **diagonales rectas** como en [la figura 5.1](#), no todos los caminos entre  $(a, b)$  y  $(m, n)$  tendría la misma longitud. En realidad, dado que se permiten algunas diagonales, la forma más rápida de llegar  $(m, n)$  de  $(a, b)$  Depende de la posición relativa entre ambos nodos:

- Si  $(m > a) \text{ o } (n > b)$ , el SP viene **marchando en diagonal siempre que sea posible** . Esto se debe a que el **objetivo**  $(m, n)$  está a la derecha de la **fuente**  $(a, b)$ . Este es el caso de  $(a = 0, b = 0)$  y  $(m = 4, n = 4)$  cuyo SP tiene sólo diagonales al igual que la longitud 4.
- Si  $(m < a) \text{ o } (n < b)$ , el SP también puede provenir de **marchar horizontal o verticalmente tanto como sea posible** . Esto se debe a que el objetivo está a la izquierda de la fuente. Este es el caso de  $(a = 2, b = 2)$  y  $(m = 0, n = 3)$  cuyo SP tiene sólo horizontales y verticales y tiene longitud 3.

[Saltar al contenido principal](#)

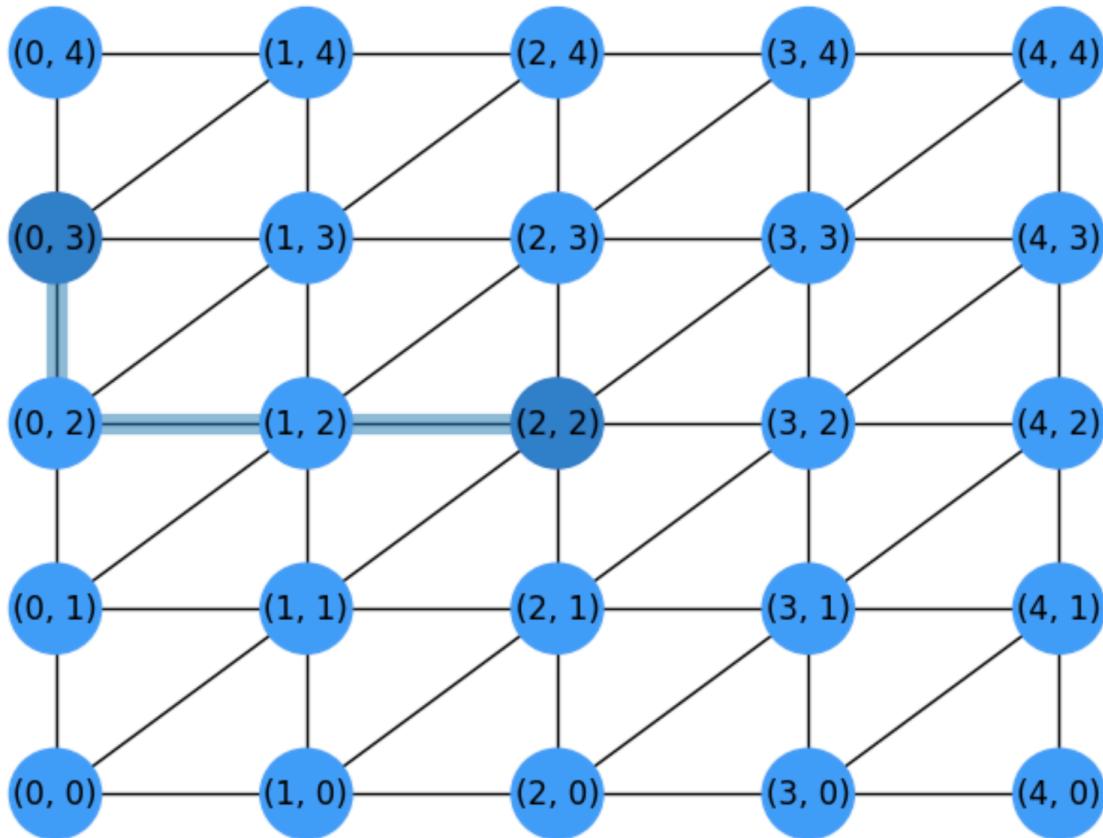


Figura 5.16-cuadrícula vecinal que permite diagonales derechas y SP.

Como no queremos considerar cada **caso particular**, es mejor sistematizar todos los casos en un **algoritmo general**. Este es el llamado **algoritmo de Dijkstra** en honor a su autor Edsger Dijkstra, quien ideó este método en 20 minutos.

### 5.1.1. Algoritmo de Dijkstra

El algoritmo, descrito en pseudocódigo en [Dijkstra](#), se puede resumir de la siguiente manera.

1. Comience en el nodo **de origen**  $s$  y visitar a sus **vecinos** para encontrar allí el **objetivo**  $t$  si es posible. Si esto sucede, la longitud del SP viene dada por el borde  $(s, t)$  y obviamente tiene longitud 1.
2. si el objetivo  $t$  no está en el barrio de la fuente, debemos **cerrar la fuente** y continuar la búsqueda en todos sus vecinos. En este caso, la longitud de SP será mayor que 1.
3. **Cerrar un nodo** significa que el algoritmo **nunca visitará este nodo más adelante**. Cerramos un nodo una vez lo seleccionamos como candidato a pertenecer al SP. El nodo de cierre se implementa eliminándolo de una lista de candidatos  $L$ .
4. Al principio, todos los nodos están incluidos en  $L$ . Esto se hace en la **fase de inicialización** del algoritmo (pasos 1 y 2).
5. Una vez que se cierra un nodo, visitamos a sus vecinos que aún están en  $L$  y comprobar si podemos alcanzar el objetivo desde ellos. Consiste en actualizar su **distancia** respecto a la fuente. Los nodos con distancias más cortas a la fuente tienen más posibilidades de ser seleccionados para cerrar y verificar si son el objetivo.
6. El algoritmo **continúa hasta que** no hay más candidatos para visitar ( $L$  está vacío) o hemos **alcanzado el objetivo** (paso 3.2).

[Saltar al contenido principal](#)

### 🔔 Algoritmo 5.1 (Dijkstra)

**Entradas** dada una red  $G = (V, E)$ , un nodo fuente  $s$  y un nodo objetivo  $t$

**Salida** Calcular una ruta de longitud mínima

```

1. para cada nodo  $v \in V$ :
    1.  $\text{dist}[v] \leftarrow \infty$ 
    2.  $\text{anterior}[v] \leftarrow -1$ 
    3. agregar  $v$  a  $L$ 
2.  $\text{dist}[s] \leftarrow 0$ 
3. mientras  $L \neq \emptyset$  :
    1.  $u \leftarrow \arg \min \text{dist}[u], u \in L$ 
    2. si  $u = t$ :
        1. romper
        3. eliminar  $u$  de  $L$ 
    4. para  $v \in \text{vecinos}(u)$ :
        1. si  $v \in L$ :
            1. nueva_dist  $\leftarrow \text{dist}[u] + w(u, v)$ 
            2. si nueva_dist <  $\text{dist}[v]$ :
                1.  $\text{dist}[v] \leftarrow \text{nueva\_dist}$ 
                2.  $\text{anterior}[v] \leftarrow u$ 
    4. devolver  $\text{dist}, \text{anterior}$ 

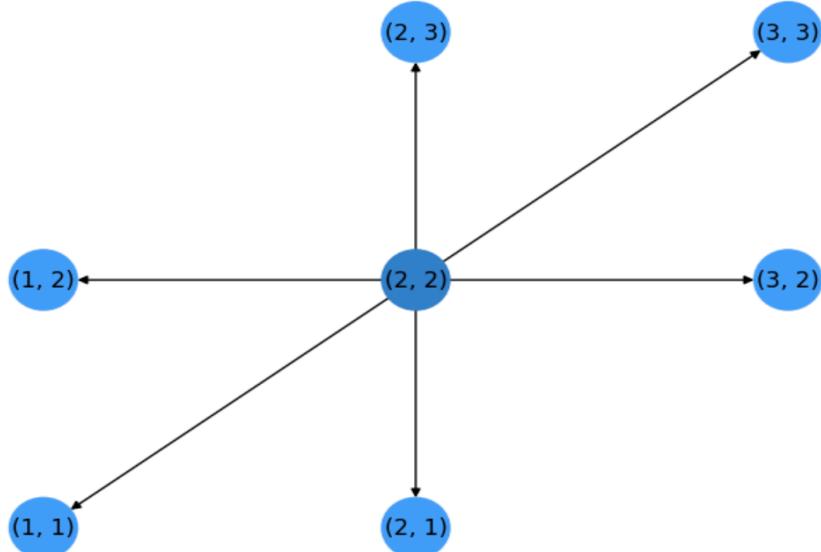
```

Fuente: [Wikipedia](#)

Mostremos cómo obtener el SP en [la Fig. 5.1](#).

#### Iteración 1

- En primer lugar, visitamos a los vecinos de la fuente  $s$  (asumimos sombreros  $\neq t$ ). Visitar a los vecinos de un nodo significa **expandirlo**.
- Para forzar tal expansión hemos establecido  $\text{dist}[s] = 0$ . Desde  $\text{dist}[v] = \infty$  para  $v \neq s$ , esto obligará a seleccionar en 3.1.
- Entonces se elimina de  $L$  (paso 3.3) porque es diferente a  $t$  y procedemos a visitar a sus vecinos  $v$  (de  $u$ ) que aún no se han ampliado (todavía en  $L$ ). Ver [Fig. 5.2](#) los 6 vecinos de  $s = (2, 2)$ . Se exploran secuencialmente (sólo los que todavía pertenecen a  $L$ ) en el bucle **for** en 4.1.
- En la figura anterior, colocamos **aristas orientadas** para mostrar quién es el nodo expandido (origen de la arista) y quiénes son sus vecinos (destinos de las aristas). ¡Esto significa que la búsqueda de SP se puede rastrear mediante un **árbol de búsqueda**!



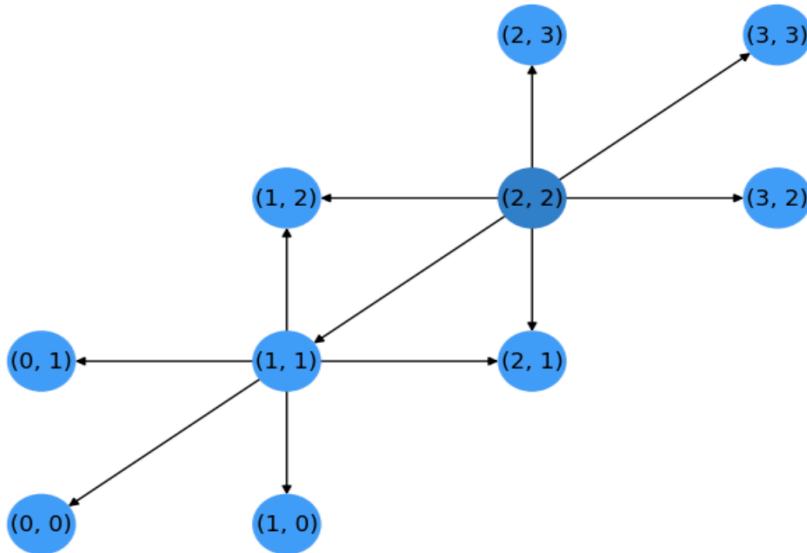
[Saltar al contenido principal](#)

Fig. 5.2 Paso 1. Ampliación del nodo fuente.

- Por lo tanto, para cada vecino  $v$  del último nodo expandido  $u = (2, 2)$  intentamos actualizar la distancia de  $v$  a la fuente. Esta distancia está codificada por  $w(u, v)$  que en principio es 1a menos que tengamos pesos de borde positivos.
- Las distancias actualizadas se almacenan en  $endist[v]$  (donde teníamos  $dist[s] = 0$  y  $dist[v] = \infty$  para  $v \neq s$ ) para que podamos elegir el **siguiente nodo candidato a expandir** que en este momento solo puede ser un vecino de  $u$ . En general, puede ser cualquier **nodo visitado** (un vecino de un nodo expandido) con la distancia más pequeña respecto a la fuente. (pueden ser muchos de ellos).
- Si un nodo visitado se convierte en **candidato para expandirse**, colocamos  $prev[v]$  el último nodo que apunta a él. En este caso, todos los vecinos de  $s$  son candidatos y están señalados por  $s$ . El único nodo que nadie ha señalado es  $s$  mismo, y es por eso que  $prev[s] = -1$  se queda para siempre.

### Iteración 2

- Al final de la primera iteración, todos los vecinos  $v$  de  $s$  satisfacen  $dist[v] = 1$  y por lo tanto son igualmente elegibles **para ser ampliados**.
- Esta es una **característica notable de Dijkstra**: el algoritmo no tiene idea de cuál de los candidatos está más cerca del objetivo. Sólo sabe a qué distancia están de  $s$ . Dijkstra **no ve el objetivo**.
- Consulte [la figura 5.3](#). Casualmente, el algoritmo elige  $(1, 1)$  en lugar de otros candidatos más alejados del objetivo  $t = (0, 3)$ .

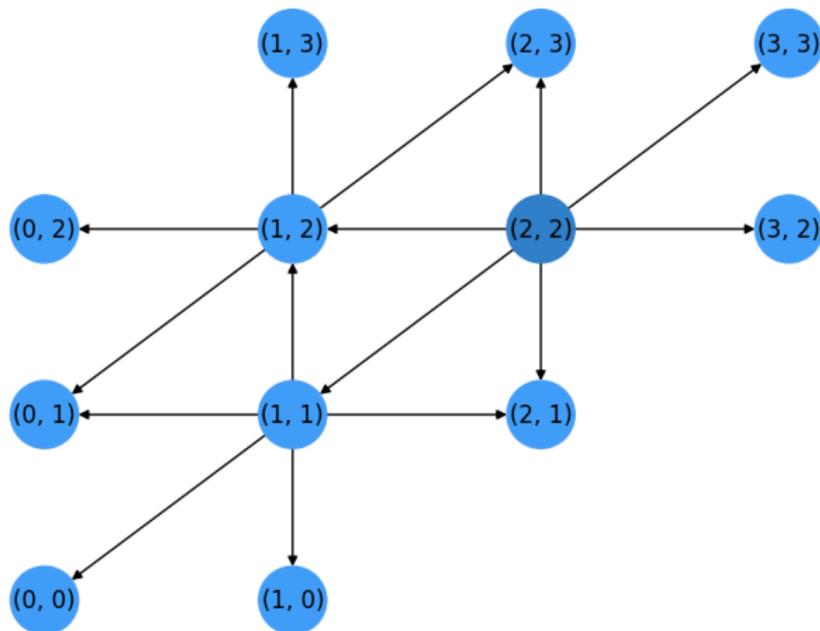
Fig. 5.3 Paso 2. Ampliación del nodo  $(1, 1)$ .

- Tenga en cuenta que algunos de los vecinos del último nodo ampliado  $u = (1, 1)$ , es decir  $(2, 1)$  y  $(1, 2)$ , aún han sido visitados por el nodo expandido anterior  $s = (2, 2)$ . Como resultado, sus entradas  $endist[v]$  no son 0 y sus distancias a la fuente no cambiarán por el momento.
- Sólo las distancias hasta la fuente de  $(0, 1)$ ,  $(0, 0)$  y  $(1, 0)$  caerán de  $\infty$  a 2. Por tanto, ninguno de estos nodos es candidato a ampliarse. Sólo los **vecinos comunes de  $(2, 2)$**  y  $(1, 1)$  ya que tienen  $dist[v] = 1 < 2$ .
- En particular, nodo  $(1, 2)$  será elegido para la expansión en la próxima iteración.

### Iteración 3

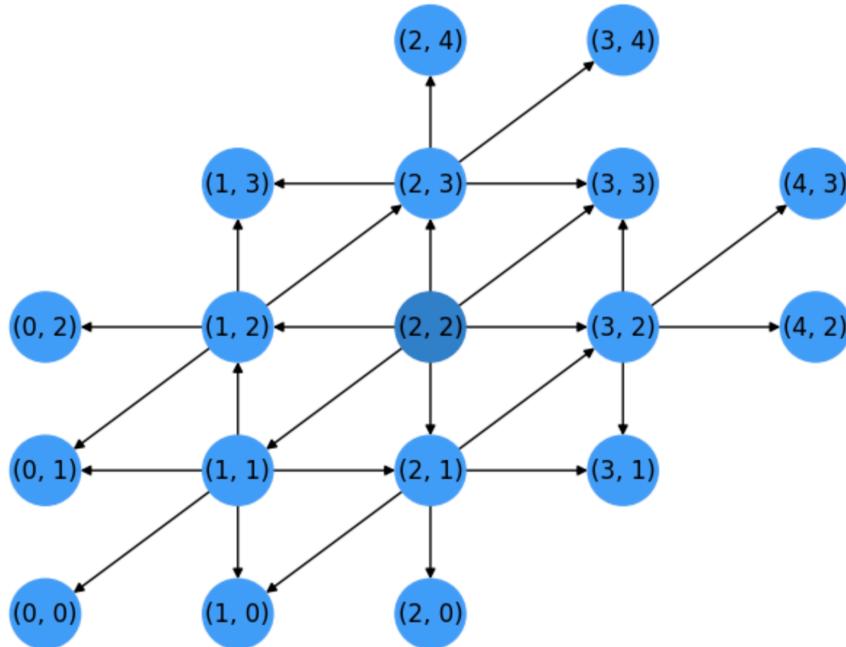
- Consulte [la figura 5.4](#). Nodou<sub>3</sub> =  $(1, 2)$  se amplía y algunos de sus vecinos,  $(2, 3)$  y  $(0, 1)$  son comunes a los nodos expandidos anteriores  $u_1 = s = (2, 2)$  y  $u_2 = (1, 1)$ .
- Dijkstra expandirá cualquier nodo no expandido de la fuente  $s = (2, 2)$  antes de considerar a otros ya que están más cerca de ello!
- En realidad, el siguiente nodo expandido es  $u_4 = (2, 1)$

[Saltar al contenido principal](#)

Fig. 5.4 Paso 3. Ampliación del nodo  $(1, 2)$ .

#### Iteración 4 y más allá

- Después de expandir  $u_4 = (2, 1)$  es el turno de  $u_5 = (2, 3)$  y luego  $u_6 = (3, 2)$ . Ver [figura 5.5](#)

Figura 5.5 Pasos 4-6. Expansión de nodos  $u_4 = (2, 1)$ ,  $u_5 = (2, 3)$ ,  $u_6 = (3, 2)$ .

- En este punto, todos los vecinos de la fuente  $s = (2, 2)$  pero  $u_7 = (3, 3)$  se han ampliado y después de expandirlo consideraremos nodos alejados de la fuente en más de 1 unidad.
- Los próximos nodos a expandirse serán:  $u_8 = (0, 0), u_9 = (0, 1), u_{10} = (0, 2)$ , todo a distancia 2. Escribe la fuente. Sin embargo el objetivo  $t = (0, 3)$ , que es vecino de  $u_{10} = (0, 2)$  pero a distancia 3 de  $s$ , no se expandirá hasta que todos los nodos a distancia 2 wrt  $s$  se amplíen:  $u_{11} = (1, 0), u_{12} = (1, 3), u_{14} = (2, 0), u_{15} = (2, 4), u_{16} = (3, 1), u_{17} = (3, 4), u_{18} = (4, 2), u_{19} = (4, 3)$  y  $u_{20} = (4, 4)$  (que es un **callejón sin salida** frente al objetivo). Ver el estado de la búsqueda después de expandir  $u_{20} = (4, 4)$  en [la figura 5.6](#):

[Saltar al contenido principal](#)

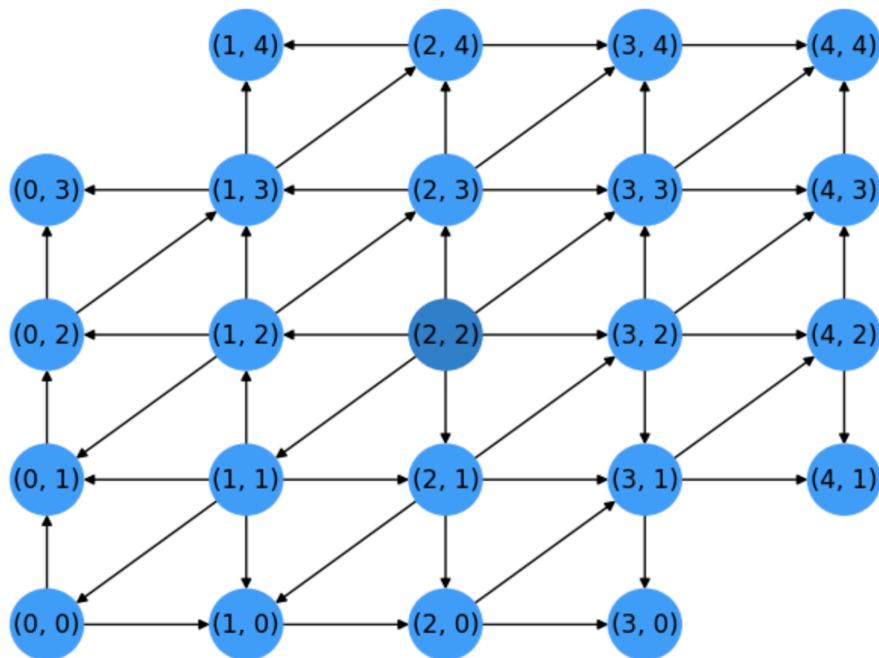


Fig. 5.6 Paso 20. Justo antes de expandir el objetivo (que aún se visita hace mucho tiempo).

- Finalmente,  $u_{21} = (0, 3) = t$  es el primer nodo a distancia 3 desde la fuente que se va a ampliar. De hecho  $(1, 4)$  o  $(0, 3)$  podría ampliarse antes, pero tuvimos **suerte y descubrimos el objetivo** en 21 pasos!

Por lo tanto, si el objetivo está a una distancia  $d$  desde la fuente, Dijkstra lo expandirá solo cuando todos los nodos a distancia  $d - 1$  de la fuente se han ampliado aún. Esta es una **búsqueda basada en el pan (BFS)**. **Ejercicio**. En la Fig. 5.7 mostramos un gráfico básico.

$H$  con el camino más corto entre la fuentes  $s = (0, 0)$  y el objetivo  $t = (2, 3)$ . Mostrar: a) el **número de iteraciones** requeridas por el algoritmo de Dijkstra así como el **árbol de búsqueda** con un  **posible orden de expansión** de los nodos. b) ¿Cuál es la **relación entre los nodos visitados y el total de nodos** en  $H$ ?

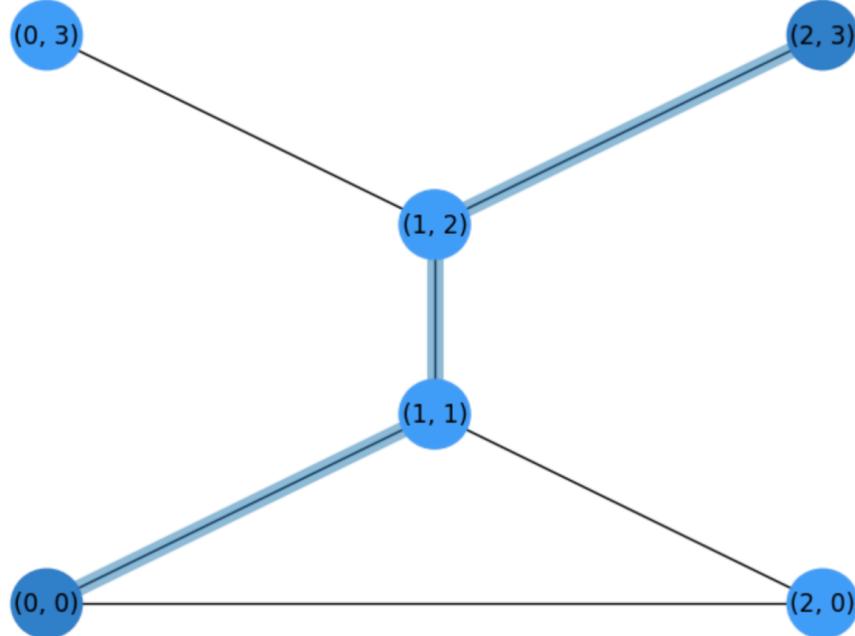
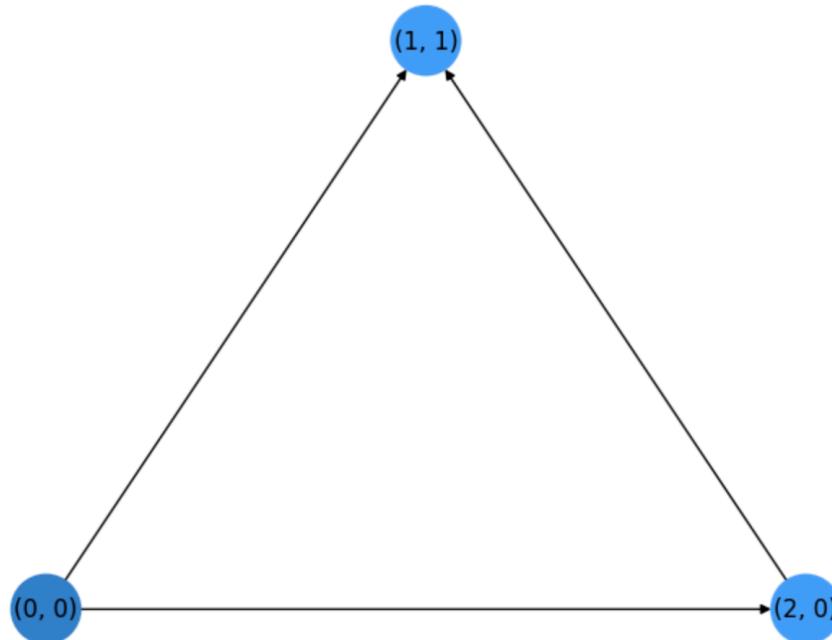


Fig. 5.7 Gráfico básico  $H$  para un ejercicio de Dijkstra.

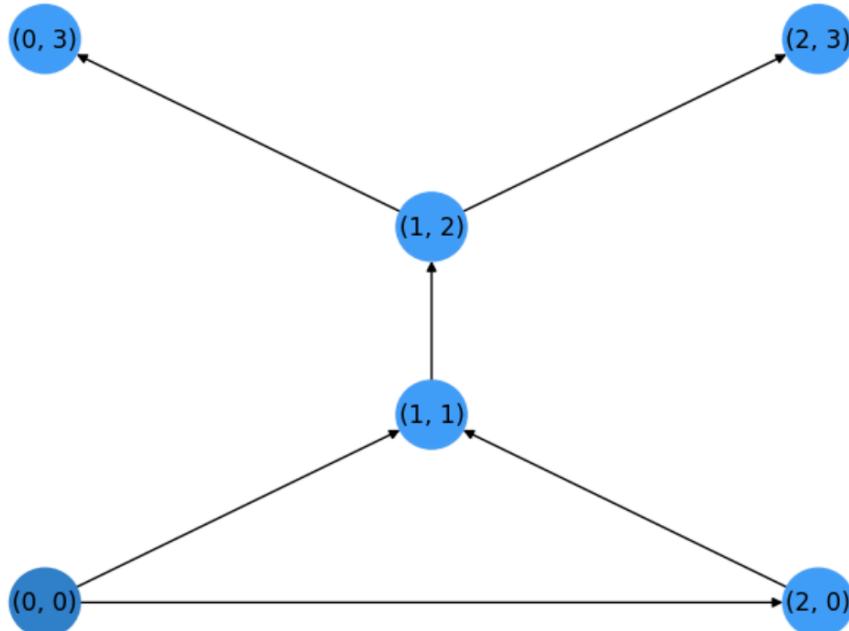
escrito a la fuentes, entonces seleccionamos  $u_2 = (2, 0)$  para la expansión. A continuación seleccionamos  $u_3 = (1, 1)$  (el otro vecino de la fuente a distancia 1). Vea el árbol de búsqueda en [la Fig. 5.8](#).



*Fig. 5.8 Árbol de búsqueda después de dos pasos (antes de expandir  $u_3 = (1, 1)$ )*

Al expandirse  $u_3 = (1, 1)$ , solo su vecino  $(1, 2)$  aún no ha sido visitado. Además, dado que todos los nodos a distancia 1 wrt la fuente se ha ampliado, ampliamos  $u_4 = (1, 2)$  a distancia 2 desde la fuente y visitar a sus vecinos  $(0, 2)$  y  $(2, 3)$ . ya que estan lejos 3, uno de ellos se expandirá una vez que verifiquemos que todos los nodos a distancia 2 se han ampliado, como es el caso.

Luego ampliamos  $u_5 = (0, 3)$  lo cual es un callejón **sin salida** para el objetivo y finalmente el objetivo  $u_6 = t = (2, 3)$ . Vea el árbol de búsqueda antes de realizar la última expansión en [la Fig. 5.9](#).



*Fig. 5.9 Árbol de búsqueda antes de expandir el objetivo.*

[Saltar al contenido principal](#)

Finalmente, como hemos expandido *todos los nodos*\* en  $H$ , la relación de expansión es 1.

**Complejidad computacional**. ¿Qué tan difícil es aplicar el algoritmo de Dijkstra al número de nodos  $|V|$  y bordes  $|E|$  en un gráfico  $G = (V, E)$ ? Para ello utilizamos la [notación Big-O](#) (complejidad de límite superior) que estudiará en breve.

- El bucle **for** para la inicialización de distancias tiene complejidad  $O(|V|)$ .
- ¿Cuántos pasos puede durar el ciclo **while**? En el peor de los casos, iteramos hasta la lista  $L$  esta vacío. Esto significa que todos los nodos se han ampliado, lo que en principio significa  $O(|V|)$  complejidad. Sin embargo, como se han expandido muchos nodos antes de expandir uno determinado, en realidad necesitamos contar los bordes del árbol de búsqueda. Como resultado, es posible que tengamos  $O(|E|)$  complejidad para gráficos **dispersos** y  $O(|E| = |V|^2)$  para **los densos**.
- Dentro del **while** hay dos bloques principales: (a) calcular el arg min  $dist[]$  y (b) expandir el nodo  $u$  y visitar a sus vecinos  $v$ .
  - Calculando el arg min  $dist[]$  Tiene una complejidad aproximada de  $O(|V|)$ .
  - La ejecución del bucle **for** para visitar a los vecinos  $u$  se puede realizar en  $O(d_{max})$ , dónde  $d_{max}$  es el grado máximo  $deg(v), v \in V$ .
- Entonces, básicamente la complejidad del bucle **while** es aproximadamente  $O(|E| \cdot |V|)$ .
- Sin embargo, al encontrar la arg min  $dist[]$  se puede optimizar para gráficos dispersos, es posible que tengamos una complejidad en el peor de los casos de  $\Theta((|E| + |V|) \log |V|)$  o mejor  $\Theta(|E| + |V| \log |V|)$ . Aquí en  $\Theta(\cdot)$  significa "en promedio".
- Sin embargo, [esta complejidad casi cuadrática](#) motiva también mejores formas algorítmicas de ir más directamente al objetivo.

## 5.1.2. Heurística y $A^*$

Los algoritmos de IA a menudo se basan en criterios **un poco más sofisticados** para expandir un nodo:

- Deja  $g(u)$  una estimación de la distancia del camino más corto **con respecto a la fuente**  $g^*(u)$ . Si el gráfico no está ponderado,  $g^*(u)$  es la longitud del camino más corto desde  $u$ .
- Ahora, presentamos  $h(u)$  como una estimación de  $h^*(u)$ , es decir, la distancia de ruta óptima (más corta) **hacia el objetivo**  $t$ . Esta función  $h(\cdot)$  se llama **heurística**.

Básicamente,  $g(u)$  estima el **trabajo aún realizado**, mientras que  $h(u)$  estima lo prometedor que es para alcanzar el objetivo  $t$ , es decir, el **trabajo a realizar**. Como resultado, el llamado [A\\*El algoritmo](#) consiste en modificar la línea 4.1.1 de Dijkstra (ver [AStar](#)):

### Algoritmo 5.2 ( $A^*$ de Dijkstra)

4. **para**  $v \in \text{vecinos}(u)$ :
1. **si**  $v \in L$ :
1.  $\text{nueva\_dist} \leftarrow g(u) + h(u)$

dónde  $g(u) = \text{dist}[u] + w(u, v)$ .

Por lo tanto  $A^*$  puede verse como una extensión de Dijkstra donde el nodo seleccionado para la expansión minimiza

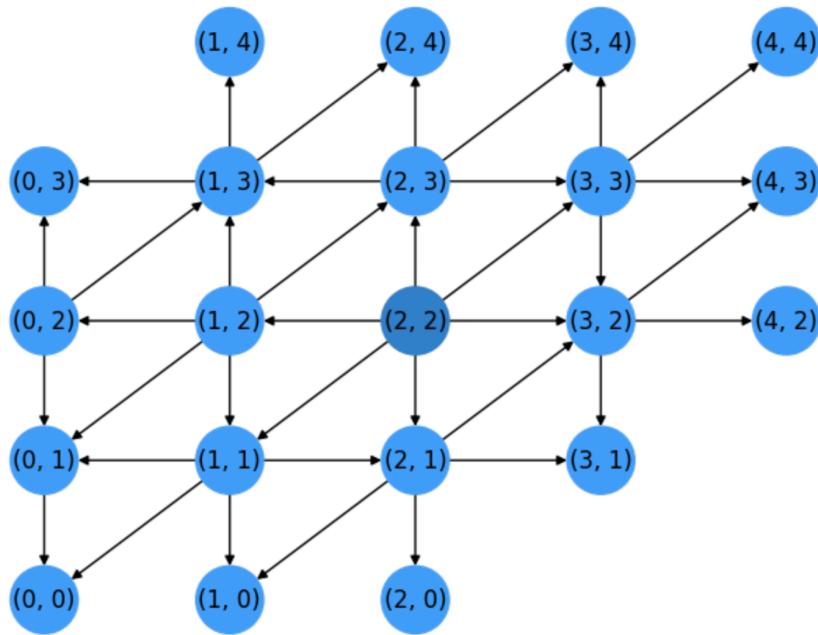
$$f(u) = g(u) + h(u).$$

**Idea principal**. La idea subyacente de  $A^*$  es que Dijkstra usa solo  $f(u) = g(u)$  pero  $A^*$  está **más informado** que Dijkstra ya que también incluye  $h(u)$ , es decir  $f(u) = g(u) + h(u)$  como medio para resolver empates.

En otras palabras, si varios nodos tienen el mismo valor  $g(u)$  y son igualmente elegibles para ser ampliados por Dijkstra,  $A^*$  selecciona  $u = \arg \min g(u) + h(u)$ .

Compárese, por ejemplo, [la Fig. 5.10](#) con [la Fig. 5.6](#). En ambos casos, tenemos  $s = (2, 2)$  y  $t = (0, 3)$ . El  $A^*$ El algoritmo procede de la siguiente manera:

[Saltar al contenido principal](#)

Fig. 5.10 Árbol de búsqueda de  $A^*$ .

- Expandir  $u_1 = s = (2, 2)$  y visitar a todos sus vecinos.
- Expandir  $u_2 = (1, 2)$  ya que es vecino de  $u_1$  más cerca del objetivo  $t$ .
- Expandir  $u_3 = (2, 3)$ , también vecino de  $u_1$ . El nodo  $(1, 3)$  no se amplía antes, a pesar de que está más cerca del objetivo porque aún no ha sido visitado (no es vecino de  $u_1$ ).
- Del mismo modo, los restantes vecinos aún no ampliados de  $u_1$  se amplían de la siguiente manera:  $u_4 = (1, 1)$ ,  $u_5 = (2, 1)$ ,  $u_6 = (3, 3)$  y  $u_7 = (3, 2)$ .
- A continuación, es el momento de expandir los nodos a distancia 2 de las fuentes, priorizando a aquellos más cercanos al objetivo  $t$ . Esto resulta en la expansión  $u_8 = (0, 2)$ ,  $u_9 = (1, 3)$  y  $u_{10} = (0, 1)$ .
- Finalmente, de ellos llegamos  $u_{11} = t = (0, 3)$ , es decir, el objetivo es la mitad de iteraciones que Dijkstra.

**¿Cómo se elige realmente la heurística?** En grillas 2D, donde cada nodo está anclado a una posición euclídea dada  $P_u = (x_u, y_u)$ , la función heurística  $h(\cdot)$  se elige según una **distanza** dada. La más obvia es la distancia **euclídea** o **aérea** entre  $P_u$  y  $P_t$  (los respectivos anclajes de  $u$  y el objetivo  $t$ ):

$$h_E(u) = \sqrt{|x_u - x_t|^2 + |y_u - y_t|^2}.$$

**Admisibilidad.** No todas las distancias son compatibles con todos los tipos de gráficos. Tal compatibilidad depende de cómo se calcula la **heurística óptima/perfecta**. Para un gráfico de cuadrícula con diagonales rectas, como el explorado anteriormente, la verdadera distancia del camino más corto  $h^*(s)$  entre  $s = (2, 2)$  y  $t = (0, 3)$  es 3. Se podría argumentar que **no lo sabemos de antemano**, y esto es cierto. En realidad, no necesitamos saberlo. Sólo necesitamos satisfacer:

$$h(u) \leq h^*(u) \quad \forall u,$$

de manera que **la A\* garantiza que encontraremos la ruta óptima/más corta**. Esta propiedad de una función heurística se llama **admisibilidad** y está garantizada por la minoración anterior, ya que **básicamente dice que nunca sobreestimamos la distancia al objetivo**. De lo contrario podríamos descartar un camino óptimo.

**Comprobación de admisibilidad.** Usando el teorema de Pitágoras podemos demostrar fácilmente que  $h(u) \leq h^*(u) \quad \forall u$ . En realidad, la distancia euclídea entre  $s = (2, 2)$  y  $t = (0, 3)$  es  $\sqrt{|0 - 2|^2 + |2 - 3|^2} = \sqrt{5} < 3$ . Puede comprobar que dicha distancia también es válida para cualquier par de nodos.

Sin embargo, considere la **distancia de Manhattan**.

para cuadrículas con diagonales rectas. Esta distancia está incluso **mejor informada** que la distancia euclíadiana ya que satisface  $h_E(s) \leq h_M(s) \leq h^*(s)$  para  $s = (2, 2)$  y  $t = (0, 3)$ , pero para  $t = (4, 4)$  tenemos eso  $h_M(s) = 4 > h^*(s) = 2$ . Por lo tanto,  $h_M(\cdot)$  no es **admisible** para este tipo de gráficos, aunque:

- La distancia de Manhattan es la elección óptima.  $h_M(u) = h^*(u)$  para 4-gráficos de barrio.
- La distancia euclíadiana es la elección óptima.  $h_E(u) = h^*(u)$  para 8-Gráficos de vecindad (incluidas las diagonales derecha e izquierda).

La distancia de Manhattan es un buen ejemplo de [Geometría del Taxi](#): si nos movemos en una ciudad real, básicamente ponemos los nodos en los cruces y no podemos atravesar los bloques de construcción. En otras palabras, **ignoramos la geometría euclíadiana** que, de hecho, tiene muchos beneficios en muchas aplicaciones de la IA.

## 5.2. Flujo máximo

### 5.2.1. Cuellos de botella, flujos y capacidades

Una caracterización esencial de un gráfico es detectar [si tiene un cuello de botella](#) y dónde se encuentra si lo hay .

Considerando aquí sólo **grafos dirigidos o dígrafos**  $G(V, E)$ , un **cuello de botella**  $B \subseteq E$  es un subconjunto  $B$  de los bordes  $E$  donde el flujo de información colapsa .

Véase, por ejemplo, [la figura 5.11](#) donde tenemos un gráfico dirigido con dos **nodos especiales**.  $s = 0$  (fuente) y  $t = 5$  (objetivo). En esta sección, los nodos de origen solo pueden **enviar información** (a sus *sucesores*), mientras que los nodos de destino (o sumideros) solo pueden **recibir información** (de sus *predecesores*).

cada borde  $e = (u, v) \in E$  está etiquetado con una **capacidad**  $c(u, v)$  lo que determina la *cantidad de información* que se puede enviar a través del borde. En este ejemplo la fuente  $s = 0$  puede enviar como *máximo* 10, digamos *fragmentos de información*. Esta cantidad se llama **flujo de borde**.  $f(u, v)$  satisface:

1. *Limitado por la capacidad*. Obviamente, el flujo a lo largo de un borde está limitado superiormente por la capacidad del borde:

$$\forall (u, v) \in E : f(u, v) \leq c(u, v) .$$

2. *Simetría sesgada*. Durante la propagación del flujo, el flujo de  $u$  a  $v$  es lo opuesto al flujo de  $v$  a  $u$ :

$$\forall (u, v) \in E : f(u, v) = -f(v, u) .$$

3. *Conservación del flujo*. El flujo total a un nodo es cero, excepto si es *fuente* que produce flujo, o *sumidero* que consume flujo:

$$\forall u \in V : u \neq s, u \neq t \Rightarrow \sum_{w \in V} f(u, w) = 0 ,$$

Por ejemplo, considere  $e = (2, 4)$  en [la figura 5.11](#). Dado que la capacidad de este borde es  $c(2, 4) = 25$ , teóricamente podríamos enviar 25 unidades de flujo de  $2 \rightarrow 4$ . Sin embargo, esto es imposible ya que:

- Nodo 2 sólo puede recibir como máximo 10 unidades de la fuente y debemos satisfacer  $f(s, 2) = f(2, 4)$ .
- Nodo 4 sólo puede recibir como máximo 16 unidades de flujo desde  $f(4, 1) + f(4, 5) \leq 16$ .

4. *Sin pérdidas*: el flujo total que sale de la fuente  $s$  debe ser igual al caudal total que llega al fregadero  $t$ :

$$\sum_{(s, u) \in E} f(s, u) = \sum_{(v, t) \in E} f(v, t) .$$

En [la figura 5.11](#), si enviamos 10 unidades de  $s = 0$  a 1 y otras 10 unidades a 2, el fregadero  $t = 5$  debe terminar de recibir 20 unidades. Si, por ejemplo, la capacidad  $c(4, t) = 5$  donde 5, esto no podría ser posible e implicaría que sólo 15 unidades se pueden producir en s.

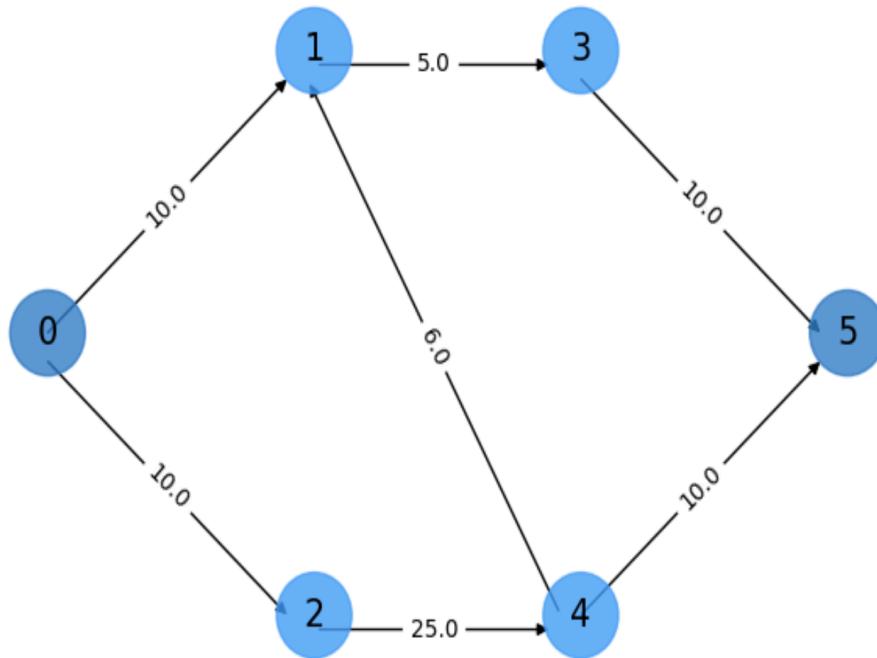


Fig. 5.11 Encuentre un cuello de botella en un gráfico dirigido.

¿Cuál es el **cuello de botella** en la figura anterior? Parece obvio que el cuello de botella está definido por los bordes en  $G(V, E)$  con menor capacidad, es decir, aquellos que pueden **limitar** la transferencia de información. Claramente, en nuestro ejemplo simple la capacidad más pequeña  $\text{esc}(1, 3) = 5$ . ¿Significa que sólo podemos producir 5 unidades en  $s$ ? Realmente no, esto sucedería si **alguno de los caminos des de  $s$  a  $t$  se ven obligados a atravesar  $e = (1, 3)$** . Este es el caso cuando el camino  $\Gamma = s \rightarrow 2 \rightarrow 4 \rightarrow t$  no existe!

En otras palabras, una arista o un conjunto de aristas es realmente un cuello de botella si restringen el flujo de información entre ellas.

En consecuencia, parece razonable explorar **los caminos entre  $s$  y  $t$**  y determinar los bordes más débiles (aquellos con capacidades más pequeñas) en relación con cada camino .

## 5.2.2. El método Ford-Fulkerson

### 5.2.2.1. El método general

Un enfoque general para encontrar el flujo máximo entre  $s$  y  $t$  en la Fig. 5.11 podría ser el siguiente:

1. encontrar un camino  $\Gamma_1$  entre  $s$  y  $t$ , por ejemplo  $\Gamma_1 = s \rightarrow 1 \rightarrow 3 \rightarrow t$ .
2. La ventaja más débil en  $\Gamma_1$  es **el cuello de botella de este camino**:  $e = (1, 3)$  y  $\text{conc}(1, 3) = 5$ . Esto significa que la máxima cantidad de información que fluye a través de  $\Gamma_1$  será  $f_1 = 5$ .
3. Asigne un flujo igual a 5 a cualquier borde en  $\Gamma_1$ . Entonces nosotros tenemos

$$f(s, 1) = f(1, 3) = f(3, t) = f_1 = 5 .$$

4. Descubramos otro camino, digamos  $\Gamma_2 = s \rightarrow 2 \rightarrow 4 \rightarrow t$ . El cuello de botella de  $\Gamma_2$  está dado por los bordes  $(s, 2)$  y  $(4, t)$  y  $\text{conc}(s, 2) = c(4, t) = 10$ . Como resultado, aunque  $c(2, 4) = 25$  solo podemos enviar  $f_2 = 10$  unidades de flujo a través de  $\Gamma_2$ .

5. Asigne un flujo igual a 10 a cualquier borde en  $\Gamma_2$ . Entonces nosotros tenemos:

$$f(s, 2) = f(2, 4) = f(4, t) = f_2 = 10 .$$

En este punto, el último camino entre  $s$  y  $t$ ,  $\Gamma_3 = s \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 3 \rightarrow t$  ya no se explora . ¿Por qué?

- como el borde  $(s, 2)$  satisface que  $c(s, 2) = f(s, 2) = f_2 = 10$  (decimos que está **saturado**) estamos seguros de que *cualquier otro camino que contenga este borde no devolverá un mejor flujo* .
- Aunque el borde  $(s, 1)$  no está saturado (supongamos que en esta sección se trata de **gráficos dirigidos**). ruta  $\Gamma_1$  contiene

[Saltar al contenido principal](#)

Entonces, el **flujo máximo total** resultante entre  $s$  y  $t$  es  $f_1 + f_2 = 5 + 10 = 15$ .

Por lo tanto, este método, llamado **método Ford-Fulkerson** (ver [Ford\\_Fulkerson](#)) se puede resumir de la siguiente manera:

1. Colocar  $i = 1$
2. **Encuentra un camino de aumento**  $\Gamma_i$ , y su(s) borde(s) de cuello de botella  $E_i$ . Este cuello de botella proporciona el flujo  $f_i$  y los correspondientes bordes saturados  $E_i$ .
3. **Actualizar los flujos** de todos los bordes en  $\Gamma_i$ .
4. Deténgase si no se pueden encontrar nuevas rutas de aumento. De lo contrario, incrementar  $i = i + 1$ .
5. **Devuelve el flujo máximo** total:  $F = \sum_i f_i$ .

#### Algoritmo 5.3 (Ford-Fulkerson)

**Entradas** dada una red  $G = (V, E)$  con capacidad de flujo  $c$ , un nodo fuente  $s$  y un nodo receptor  $t$

**Salida** Calcular un flujo  $f$  deseado de valor máximo

1.  $f(u, v) \leftarrow 0$  para todos los bordes  $(u, v)$
2. **mientras** hay un camino  $\Gamma$  deseado en  $G_f$  tal que  $c_f(u, v) > 0$  para todos los bordes  $(u, v) \in \Gamma$ :
  1. Encontrar  $c_f(\Gamma) = \min\{c_f(u, v) : (u, v) \in \Gamma\}$
  2. **para** cada borde  $(u, v) \in \Gamma$ 
    1.  $f(u, v) \leftarrow f(u, v) + c_f(\Gamma)$  (*Enviar flujo a lo largo del camino*)
    2.  $f(v, u) \leftarrow f(v, u) - c_f(\Gamma)$  (*El flujo podría "devolverse" más tarde*)
  3. **devolver**  $f$ .

### 5.2.2.2. Rutas de aumento y gráfico

Algunos elementos interesantes del algoritmo anterior son los siguientes.

**Caminos de aumento**. Un concepto central en este método es los  $s-t$  caminos (caminos dirigidos entre  $s$  y  $t$ ), propuesto en cada iteración. Se denominan *caminos aumentantes* porque no contienen ningún borde saturado.

**Gráfico residual**  $G_f = (V, E_f, c_f, f)$  es el gráfico donde se actualizan los flujos. Funciona de la siguiente manera:

- $E_f$  tiene  $2|E|$  bordes, es decir, si  $(u, v) \in E$  entonces incluimos **ambos**  $(u, v)$  y  $(v, u)$  en  $E_f$ .
- Inicialmente todos los bordes  $(u, v) \in E_f$  tienen  $f(u, v) = 0$  (Paso 1)
- La **capacidad residual**  $c_f(u, v) \in E_f$  siempre viene dado por

$$c_f(u, v) = c(u, v) - f(u, v) \geq 0 .$$

- En el gráfico residual inicial  $G_f$  (ver [Fig. 5.12](#)) tenemos  $c_f(u, v) = c(u, v)$  para  $(u, v) \in E$  y  $c_f(u, v) = 0$  de lo contrario.

[Saltar al contenido principal](#)

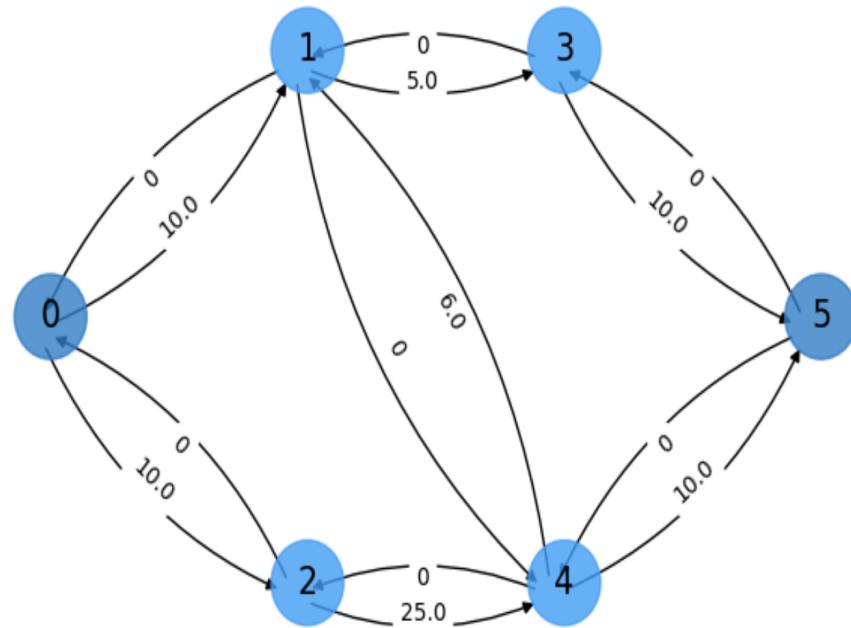


Fig. 5.12 Gráfico residual inicial.

- El gráfico residual es una **estructura de datos auxiliar** para realizar un seguimiento de la capacidad actual después de cada actualización de los flujos.
- Además, el gráfico residual registra los **flujos de corriente**, que siempre cumplen las propiedades de flujo anteriores, en particular  $f(u, v) = -f(v, u)$  para  $(u, v) \in E_f$ .
- Finalmente, es muy importante tener en cuenta que **todos los bordes en  $E_f$**  se puede utilizar para encontrar rutas de **aumento**, como vemos a continuación.

En [la figura 5.13](#), mostramos los **flujos finales** de  $G_f$  para el ejemplo anterior. ¿Puedes interpretar los resultados?

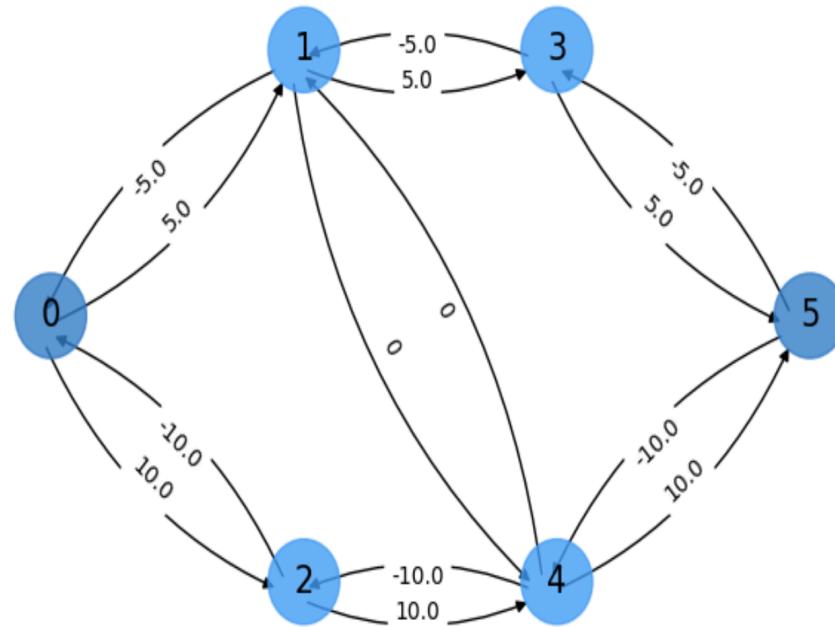


Fig. 5.13 Flujos finales del gráfico residual.

Podemos ver que los dos caminos aumentados descubiertos por el algoritmo son

$$\Gamma_1 = s \rightarrow 1 \rightarrow 3 \rightarrow t$$

[Saltar al contenido principal](#)

con flujos respectivos  $f_1 = 5$  y  $f_2 = 10$ . Lo sabemos por la figura anterior porque:

- Los dos flujos que salen de son respectivamente  $f_1 = 5$  y  $f_2 = 10$ , exactamente los mismos flujos que llegan a  $t$ .
- Borde  $(4, 1)$  se omite (no aparece en ninguna ruta de aumento) porque ambos bordes  $(4, 1)$  y  $(1, 4)$  tienen flujo nulo!

**¿Cómo encontramos los caminos aumentantes?** Como resultado del razonamiento anterior, todo lo que necesitamos para resolver el problema del flujo máximo es elegir adecuadamente las rutas de aumento.  $\Gamma_i$ . En realidad, el algoritmo para hacerlo no está especificado en el método Ford-Fulkerson. En realidad, podemos enviar un caminante aleatorio desde  $s$  en cada iteración. Sin embargo, hacerlo puede aumentar la complejidad del método.

Ejemplifiquemos esto con la siguiente ejecución:

1. Utilizar el  $G_f$  de [la figura 5.12](#).
2. Establecer todos los flujos a cero  $f(u, v) = 0, \forall (u, v) \in E_f$ .
3. Imagine, por ejemplo, que el primer camino de aumento propuesto por un caminante aleatorio es

$$\Gamma_1 = s \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 3 \rightarrow t .$$

Mirando la [figura 5.11](#), el cuello de botella de  $\Gamma_1$  está dado por el borde  $(1, 3)$  con  $c_f(1, 3) = 5$ . Como resultado, a todos los bordes de este camino se les asigna un flujo  $f_1 = 5$ . El borde  $(s, 2)$  está claramente **insaturado** y, por tanto, se puede utilizar en rutas posteriores.

4. Por ejemplo, supongamos que el segundo camino aumentado propuesto por un caminante aleatorio es **bastante más corto** (en términos del número de aristas) que  $\Gamma_1$ :

$$\Gamma_2 = s \rightarrow 1 \rightarrow 4 \rightarrow t .$$

**Recuerde :** podemos hacer eso porque todos los bordes en  $E_f$  se puede utilizar para encontrar caminos de aumento!

Los flujos de corriente de las aristas en este camino son:

$$\begin{aligned} f(s, 1) &= 0 \text{ with residual capacity } c_f(s, 1) = 10 - 0 = 10 \\ f(1, 4) &= -5 \text{ with residual capacity } c_f(1, 4) = 0 - (-5) = 5 \\ f(4, t) &= 0 \text{ with residual capacity } c_f(4, t) = 10 - 0 = 10 \end{aligned}$$

y los respectivos valores negativos para los bordes inversos. Los flujos de corriente en los otros bordes son:

$$\begin{aligned} f(s, 2) &= 5 \text{ with residual capacity } c_f(s, 2) = 10 - 5 = 5 \\ f(2, 4) &= 5 \text{ with residual capacity } c_f(2, 4) = 10 - 5 = 5 \\ f(4, 1) &= 5 \text{ with residual capacity } c_f(4, 1) = 6 - 5 = 1 \\ f(1, 3) &= 5 \text{ with residual capacity } c_f(1, 3) = 5 - 5 = 0 \\ f(3, 5) &= 5 \text{ with residual capacity } c_f(3, 5) = 10 - 5 = 5 \\ f(s, 1) &= f(1, s) = 0 \text{ with residual capacity } c_f = 10 - 0 = 10 \end{aligned}$$

respectivos valores negativos para los bordes inversos.

Desde el cuello de botella de  $\Gamma_2 = s \rightarrow 1 \rightarrow 4 \rightarrow t$  viene dado (de acuerdo con el Paso 2.1) por

$$f_2 = \min(c_f(s, 1), c_f(1, 4), c_f(4, t)) = \min(10, 5, 10) = 5$$

Actualizaremos los flujos en los bordes de  $\Gamma_2$  (Paso 2.2) de la siguiente manera:

$$\begin{aligned} f(s, 1) &= 0 + 5 = 5 \text{ whereas the reverse flow becomes } f(1, s) = 0 - 5 = -5 \\ f(1, 4) &= -5 + 5 = 0 \text{ whereas the reverse flow becomes } f(4, 1) = 5 - 5 = 0 \\ f(4, t) &= 0 + 5 = 5 \text{ whereas the reverse flow becomes } f(t, 4) = 0 - 5 = -5 . \end{aligned}$$

Aquí **hemos recuperado** el flujo  $f(1, 4)$  que aumenta de  $-5$  a  $0$ . Por lo tanto, los bordes  $(1, 4)$  y  $(4, 1)$  todavía están insaturados ya que la capacidad es  $6$ .

[Saltar al contenido principal](#)

**Por tanto** : el hecho de que en el resultado final aparezca una arista con caudal nulo ocurre si ha formado parte de un camino aumentado y luego de otro en sentido inverso que ha **restablecido el caudal** . Este es realmente **el propósito del Paso 2.2** .

### 5.2.2.3. Trampas subóptimas

Después del segundo paso, observe que el flujo total es  $f_1 + f_2 = 5 + 5 = 10$  lo cual no es óptimo en absoluto (sabemos que el flujo máximo para este ejemplo es 15).

En realidad, este camino  $\Gamma_2 = s \rightarrow 1 \rightarrow 4 \rightarrow t$  se puede volver a tomar. ¿Qué pasa si  $\Gamma_3 = \Gamma_2$ ?

- En primer lugar, los flujos y las capacidades residuales actuales de los bordes en este camino son

$$\begin{aligned}f(s, 1) &= 5 \text{ with residual capacity } c_f(s, 1) = 10 - 5 = 5 \\f(1, 4) &= 0 \text{ with residual capacity } c_f(1, 4) = 6 - 0 = 6 \\f(4, t) &= 5 \text{ with residual capacity } c_f(4, t) = 10 - 5 = 5\end{aligned}$$

- En segundo lugar, el obstáculo de este camino es

$$f_3 = \min(c_f(s, 1), c_f(1, 4), c_f(4, t)) = \min(5, 6, 5) = 5.$$

- Entonces, los nuevos flujos para estos bordes son:

$$\begin{aligned}f(s, 1) &= 5 + 5 = 10 \text{ whereas the reverse flow becomes } f(1, s) = -5 - 5 = -10 \\f(1, 4) &= 0 + 5 = 5 \text{ whereas the reverse flow becomes } f(4, 1) = 0 - 5 = -5 \\f(4, t) &= 5 + 5 = 10 \text{ whereas the reverse flow becomes } f(t, 4) = -5 - 5 = -10.\end{aligned}$$

- De esta manera, el flujo total es  $f_1 + f_3 = 5 + 5 = 10$ , todavía subóptimo. Sin embargo, hemos saturado ambos bordes. ( $s, 1$ ) y ( $4, t$ ) y este camino  $\Gamma_3$  no se puede volver a utilizar.
- En lugar de eso, descubriremos el camino.

$$\Gamma_4 = s \rightarrow 2 \rightarrow 4 \rightarrow t,$$

que proporcionará el flujo máximo óptimo:  $f_1 + f_4 = 5 + 10 = 15$ .

- En otras palabras, si tenemos la mala suerte de elegir un camino cuyos bordes no están saturados, pero el recorrido máximo a través de estos bordes es, digamos  $f$ , podemos reutilizar este camino hasta que el flujo de uno de sus bordes alcance  $f$  y se satura.
- Como podemos visitar potencialmente todos los bordes de  $E_f$  (si el camino no saturado es muy largo), la complejidad de este método es  $O(f \cdot |E|)$ , es decir, ¡depende del valor del caudal máximo que recorre el gráfico!
- Para comprender plenamente esta complejidad proponemos el siguiente ejercicio sencillo.

**Ejercicio** . En [la figura 5.14](#) mostramos un dígrafo básico.  $G$  con  $s = 0$  y  $t = 3$ , que ilustra el peor caso para el método Ford-Fulkerson. Sucede si seleccionamos alternativamente los siguientes caminos

$$\begin{aligned}\Gamma_{odd} &= s \rightarrow 1 \rightarrow 2 \rightarrow t \\ \Gamma_{even} &= s \rightarrow 2 \rightarrow 1 \rightarrow t\end{aligned}$$

[Saltar al contenido principal](#)

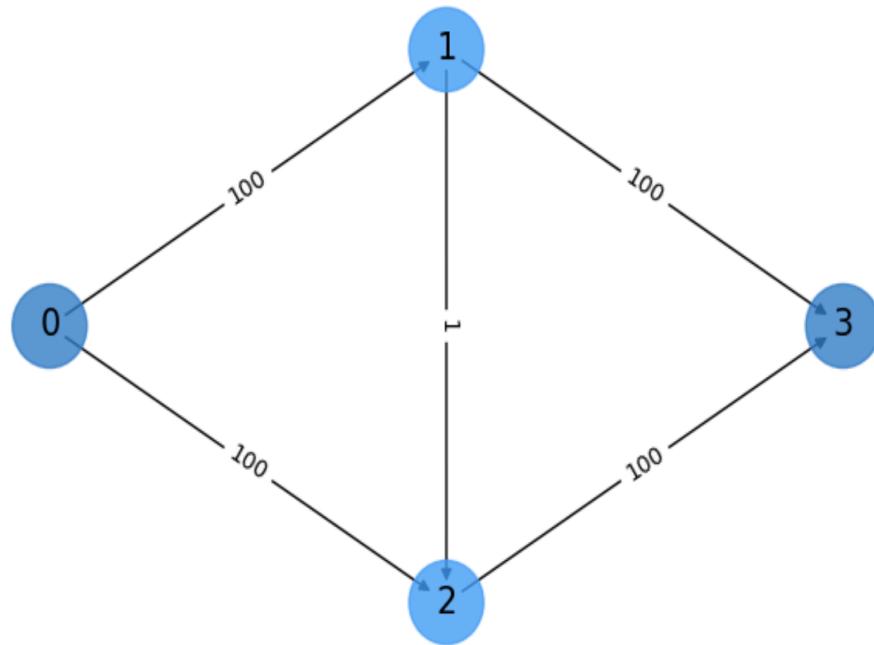
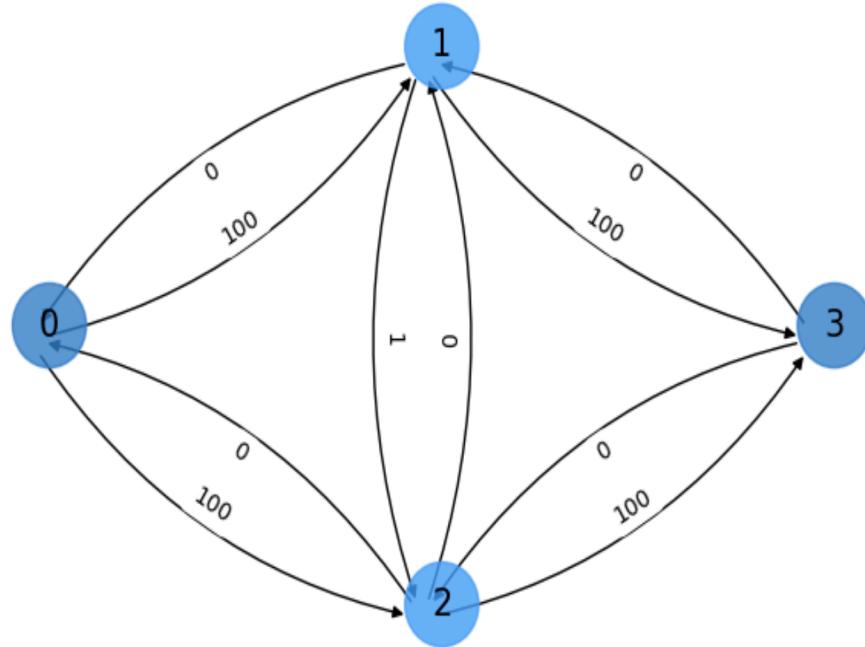


Fig. 5.14 Ilustrando el peor caso de Ford-Fulkerson.

Recuerda eso  $\Gamma_{even}$  se puede seleccionar porque buscamos rutas en el gráfico residual. El gráfico residual inicial  $G_f$  se muestra en la figura 5.15

Fig. 5.15 Gráfico residual inicial  $G_f$  del ejercicio.

**¿Por qué éste es el peor caso del método Ford-Fulkerson? Iteración 1 :** Después de configurar todos los flujos en

0, seleccionamos la ruta  $\Gamma_{odd} = s \rightarrow 1 \rightarrow 2 \rightarrow t$ . Obviamente, el cuello de botella viene dado por  $c_f(1, 2) = 1$ . Los flujos resultantes de los bordes en  $\Gamma_{odd}$  sus capacidades residuales son:

[Saltar al contenido principal](#)

$f(s, 1) = 0 + 1 = 1$  with residual capacity  $c_f(s, 1) = 100 - 1 = 99$   
 $f(1, s) = 0 - 1 = -1$  with residual capacity  $c_f(1, s) = 0 - (-1) = 1$   
**f(1, 2) = 0 + 1 = 1** with residual capacity **c<sub>f</sub>(1, 2) = 1 - 1 = 0**  
 $f(2, 1) = 0 - 1 = -1$  with residual capacity  $c_f(2, 1) = 0 - (-1) = 1$   
 $f(2, t) = 0 + 1 = 1$  with residual capacity  $c_f(2, t) = 100 - 1 = 99$   
 $f(t, 2) = 0 - 1 = -1$  with residual capacity  $c_f(t, 2) = 0 - (-1) = 1$

Como resultado, borde(1, 2) está saturado y no podemos utilizar el camino  $\Gamma_{even}$  por el momento.

**Iteración 2 :** usemos entonces

$\Gamma_{even} = s \rightarrow 2 \rightarrow 1 \rightarrow t$ , porque  $c_f(2, 1) = 1 > 0$ . El cuello de botella de este camino en realidad viene dado por  $c_f(2, 1) = 1$ . Los flujos resultantes de los bordes en  $\Gamma_{even}$  y sus capacidades residuales son:

$f(s, 2) = 0 + 1 = 1$  with residual capacity  $c_f(s, 2) = 100 - 1 = 99$   
 $f(2, s) = 0 - 1 = -1$  with residual capacity  $c_f(2, s) = 0 - (-1) = 1$   
**f(2, 1) = -1 + 1 = 0** with residual capacity **c<sub>f</sub>(2, 1) = 0 - 0 = 0**  
 $f(1, 2) = 1 - 1 = 0$  with residual capacity  $c_f(1, 2) = 1 - 0 = 1$   
 $f(1, t) = 0 + 1 = 1$  with residual capacity  $c_f(1, t) = 100 - 1 = 99$   
 $f(t, 1) = 0 - 1 = -1$  with residual capacity  $c_f(t, 1) = 0 - (-1) = 1$

**Iteración 3 :** No podemos usar  $\Gamma_{even} = s \rightarrow 2 \rightarrow 1 \rightarrow t$ , porque  $c_f(2, 1) = 0$ , pero podemos usar de nuevo  $\Gamma_{odd} = s \rightarrow 1 \rightarrow 2 \rightarrow t$  desde  $c_f(1, 2) = 1 > 0$ . Al hacerlo, tenemos nuevamente un cuello de botella de 1:

$f(s, 1) = 1 + 1 = 2$  with residual capacity  $c_f(s, 1) = 100 - 2 = 98$   
 $f(1, s) = -1 - 1 = -2$  with residual capacity  $c_f(1, s) = 0 - (-2) = 2$   
**f(1, 2) = 0 + 1 = 1** with residual capacity **c<sub>f</sub>(1, 2) = 1 - 1 = 0**  
 $f(2, 1) = 0 - 1 = -1$  with residual capacity  $c_f(2, 1) = 0 - (-1) = 1$   
 $f(2, t) = 1 + 1 = 2$  with residual capacity  $c_f(2, t) = 100 - 2 = 98$   
 $f(t, 2) = -1 - 1 = -2$  with residual capacity  $c_f(t, 2) = 0 - (-2) = 2$

Después de tres iteraciones, tenemos suficiente evidencia para demostrar que:

- (a) **Cada vez que ejecutamos**(1, 2) empujamos una unidad adicional de flujo a través de él, aumentando así el flujo  $f(s, 1)$ , y como resultado saturamos este borde(1, 2) y habilitamos el borde(2, 1).
- (b) Entonces, **si elegimos atravesar**(2, 1), aumentamos el flujo  $f(s, 2)$  y como resultado nos saturamos(2, 1) y habilitamos(1, 2)

Por lo tanto, después  $2 \cdot 100$  iteraciones, somos capaces de saturar ambos  $\Gamma_{odd}$  y  $\Gamma_{even}$ , enviando así 200 unidades. Dado que cada iteración toma casi todas las aristas en  $E$ , la complejidad del tiempo si es de orden  $O(f \cdot |E|)$ .

El **mensaje principal** del ejercicio anterior es mostrar que, **en ausencia de una forma sistemática (no aleatoria) de elegir los caminos**, el método Ford-Fulkerson tiene una complejidad que depende de los valores del flujo. Si nuestros flujos son como los del ejemplo (tenemos un enlace *con fuga*(1, 2) con un flujo muy pequeño), esta elección de camino conduce al peor caso del método.

### 5.2.3. El método Edmonds-

Este método básicamente aborda exactamente el punto de **incorporar una forma sistemática de explorar los caminos aumentantes**. Básicamente consiste en incorporar un método BFS (Breadth-First-Search), como el algoritmo de Dijkstra, en el Paso 2 del método Ford-Fulkerson (ver [Edmonds\\_Karp](#) ).

### 🔔 Algoritmo 5.4 (Edmonds-Karp)

**Entradas** dada una red  $G = (V, E)$  con capacidad de flujo  $c_f$ , un nodo fuente  $s$  y un nodo receptor  $t$

**Salida** Calcular un flujo  $f$  de  $s$  a  $t$  de valor máximo

1.  $f(u, v) \leftarrow 0$  para todos los bordes  $(u, v)$
2. **mientras** hay un **camino más corto** BFS  $\Gamma$  de  $s$  a  $t$  en  $G_f$  tal que  $c_f(u, v) > 0$  para todos los bordes  $(u, v) \in \Gamma$ :

  1. Encontrar  $c_f(\Gamma) = \min\{c_f(u, v) : (u, v) \in \Gamma\}$
  2. **para** cada borde  $(u, v) \in \Gamma$ 
    1.  $f(u, v) \leftarrow f(u, v) + c_f(\Gamma)$  (*Enviar flujo a lo largo del camino*)
    2.  $f(v, u) \leftarrow f(v, u) - c_f(\Gamma)$  (*El flujo podría "devolverse" más tarde*)
  3. **devolver**  $f$ .

En este punto es necesario hacer algunas precisiones:

- El BFS busca los **caminos más cortos** (en términos de número de aristas) entre  $s$  y  $t$  en el gráfico residual.
- Los caminos más cortos son más interesantes que los caminantes aleatorios o incluso DFS (búsqueda primero en profundidad) porque cuanto **más largo es el camino, más probable es encontrar un borde de fuga** (un borde con una capacidad muy pequeña frente a otros) en él. Los caminos largos suelen dar como resultado el peor caso del algoritmo.

En realidad, el peor caso de Ford-Fulkerson ilustrado en [la figura 5.15](#) se evita fácilmente encontrando dos caminos:

$$\Gamma_1 = s \rightarrow 1 \rightarrow t$$

y

$$\Gamma_2 = s \rightarrow 2 \rightarrow t$$

evitando así el borde del cuello de botella  $(1, 2)$  (ver [Fig. 5.16](#), donde el flujo total es  $100 + 100 = 200$ ).

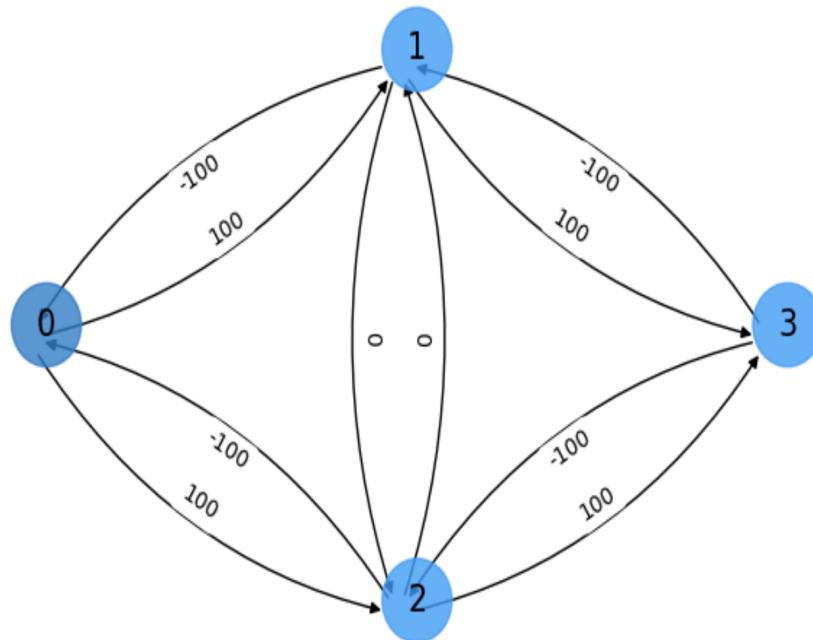


Fig. 5.16 Gráfico residual final  $G_f$  del ejercicio.

**Por lo tanto**: el hecho de que aparezca una arista con flujo cero en el resultado final **también ocurre** cuando evitamos esta arista (recordemos que el flujo está puesto en 0 inicialmente para todos los bordes)..

**Complejidad**. En cuanto a la complejidad de Edmonds-Karp (EK), ya que cada iteración requiere ejecutar un algoritmo BFS como Dijkstra, con complejidad  $O(|V| \cdot |E|)$  uno podría esperar esta complejidad para EK. Sin embargo:

[Saltar al contenido principal](#)

- En cada iteración del algoritmo, la ruta más corta entre la fuente y todos los demás vértices en el gráfico residual debe volverse monótona. Es decir, siempre es creciente o no decreciente. Como resultado, la trayectoria en el gráfico real está disminuyendo. Esto limita la longitud de una iteración de Edmonds-Karp a  $O(|E|)$ .
- Ninguna modificación del gráfico residual reduce jamás la distancia desde la fuente a cualquier vértice; Con esto, y el hecho de que BFS siempre encuentra el camino más corto, es posible demostrar que después de encontrar como máximo  $|V| \cdot |E|$  Al aumentar las rutas, no queda ninguna ruta de origen a sumidero en el gráfico residual.
- Entonces, la complejidad total es  $O(|V| \cdot |E|^2)$ .

**Ejercicio** . En la figura 5.17 mostramos un dígrafo.  $G_{\text{cons}} = 0$  y  $t = 10$ , que ilustra el caso general del método EK. Supongamos que se le proporcionan las rutas de aumento descubiertas por el algoritmo BFS. En este caso hemos utilizado el **BFS bidireccional** explotado por Networkx. Tenemos, de todos modos, los caminos:

$$\Gamma_1 = s \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow t$$

$$\Gamma_2 = s \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow t$$

$$\Gamma_3 = s \rightarrow 3 \rightarrow 6 \rightarrow 8 \rightarrow t$$

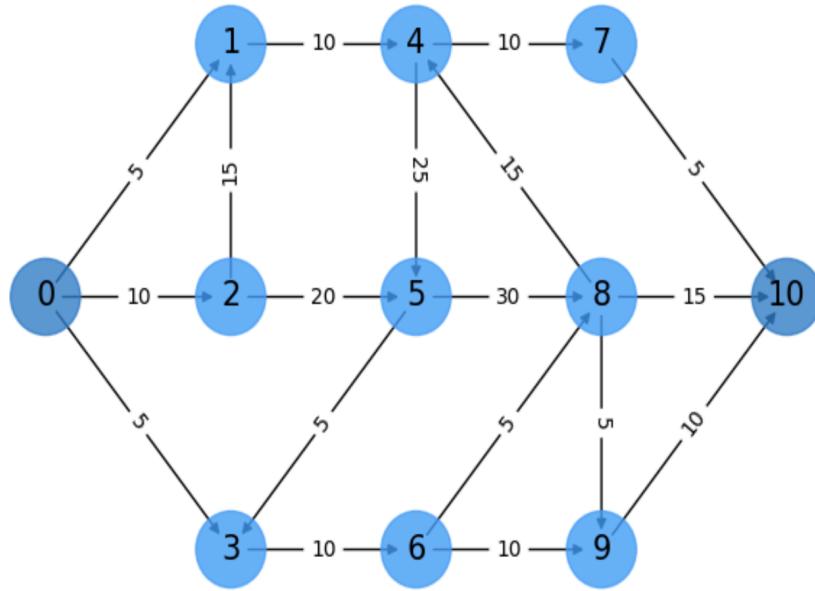


Fig. 5.17 Un ejemplo de EK más complejo.

El ejercicio consiste en explicar cómo contribuyen estos caminos a encontrar el flujo máximo y qué es Iteración 1 . Después de inicializar todos los flujos en el gráfico residual como

$f = 0$ , comenzemos por  $\Gamma_1 = s \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow t$ . Su cuello de botella está al borde  $e = (s, 1)$  ( $e = (7, t)$ ) donde tenemos  $c(e) = 5$ . De todos modos, esto conduce a los siguientes flujos y capacidades residuales:

$$\begin{aligned} f(s, 1) &= 0 + 5 = 5 \text{ with residual capacity } c_f(s, 1) = 5 - 5 = 0 \\ f(1, s) &= 0 - 5 = -5 \text{ with residual capacity } c_f(1, s) = 0 - (-5) = 5 \\ f(1, 4) &= 0 + 5 = 5 \text{ with residual capacity } c_f(1, 2) = 10 - 5 = 5 \\ f(4, 1) &= 0 - 5 = -5 \text{ with residual capacity } c_f(2, 1) = 0 - (-5) = 5 \\ f(4, 7) &= 0 + 5 = 5 \text{ with residual capacity } c_f(4, 7) = 10 - 5 = 5 \\ f(7, 4) &= 0 - 5 = -5 \text{ with residual capacity } c_f(4, 7) = 0 - (-5) = 5 \\ f(7, t) &= 0 + 5 = 5 \text{ with residual capacity } c_f(7, 10) = 5 - 5 = 0 \\ f(t, 7) &= 0 - 5 = -5 \text{ with residual capacity } c_f(t, 7) = 0 - (-5) = 5 \end{aligned}$$

Esto da como resultado la saturación de los bordes del cuello de botella  $(s, 1)$  y  $(7, t)$ . Estos dos bordes no se pueden utilizar

[Saltar al contenido principal](#)

tomar un camino con el borde saturado

$(t, 1)$  el algoritmo elige otro camino más corto **en el gráfico residual**. En este caso, se necesita  $\Gamma_2 = s \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow t$  también con longitud  $|\Gamma_2| = 4$  como lo ha hecho  $\Gamma_1$ . El cuello de botella viene dado por el borde  $e = (t, 2)$  con capacidad  $c(t, 2) = 10$ . A continuación se procede a la actualización de caudales y capacidades residuales:

$$\begin{aligned} f(s, 2) &= 0 + 10 = 10 \text{ with residual capacity } c_f(s, 2) = 10 - 10 = 0 \\ f(2, s) &= 0 - 10 = -10 \text{ with residual capacity } c_f(2, s) = 0 - (-10) = 10 \\ f(2, 5) &= 0 + 10 = 10 \text{ with residual capacity } c_f(1, 2) = 20 - 5 = 10 \\ f(5, 2) &= 0 - 10 = -10 \text{ with residual capacity } c_f(5, 2) = 0 - (-10) = 10 \\ f(5, 8) &= 0 + 10 = 10 \text{ with residual capacity } c_f(5, 8) = 30 - 10 = 20 \\ f(8, 5) &= 0 - 10 = -10 \text{ with residual capacity } c_f(8, 5) = 0 - (-10) = 10 \\ f(8, t) &= 0 + 10 = 10 \text{ with residual capacity } c_f(8, 10) = 15 - 10 = 5 \\ f(t, 8) &= 0 - 10 = -10 \text{ with residual capacity } c_f(t, 8) = 0 - (-10) = 10 \end{aligned}$$

Esto da como resultado la saturación del borde  $(t, 1)$  y por lo tanto no podemos usarlo en ningún otro camino. En este punto, **tenga en cuenta que cada camino de aumento descarta muchos otros en el futuro** y es por eso que esto motiva la complejidad computacional descrita anteriormente. **Iteración 3**. Las vías aumentadas restantes no pueden utilizar ni el borde

$(s, 1)$  ni el borde  $(s, 2)$ . En consecuencia, deben apartarse de  $(s, 3)$ . Algunos de ellos tienen longitud 4 (por favor identifíquelo) pero el algoritmo elige  $\Gamma_3 = s \rightarrow 3 \rightarrow 6 \rightarrow 8 \rightarrow t$  también con longitud  $|\Gamma_3| = 4$ . Su cuello de botella son los bordes.

$e = (t, 3)$  y  $e = (6, 8)$  con capacidades  $c(e) = 5$ . Esto da como resultado los siguientes caudales y capacidades residuales.

$$\begin{aligned} f(s, 3) &= 0 + 5 = 5 \text{ with residual capacity } c_f(s, 3) = 5 - 5 = 0 \\ f(3, s) &= 0 - 5 = -5 \text{ with residual capacity } c_f(3, s) = 0 - (-5) = 5 \\ f(3, 6) &= 0 + 5 = 5 \text{ with residual capacity } c_f(3, 6) = 10 - 5 = 5 \\ f(6, 3) &= 0 - 5 = -5 \text{ with residual capacity } c_f(6, 3) = 0 - (-5) = 5 \\ f(6, 8) &= 0 + 5 = 5 \text{ with residual capacity } c_f(6, 8) = 5 - 5 = 0 \\ f(8, 6) &= 0 - 5 = -5 \text{ with residual capacity } c_f(8, 6) = 0 - (-5) = 5 \\ f(8, t) &= 10 + 5 = 15 \text{ with residual capacity } c_f(8, t) = 10 - 5 = 5 \\ f(t, 8) &= -10 - 5 = -15 \text{ with residual capacity } c_f(t, 8) = 0 - (-15) = 15 \end{aligned}$$

**Flujo total**. Desde que tenemos  $f_1 = 5, f_2 = 10, y f_3 = 5$ , el flujo total es  $f_1 + f_2 + f_3 = 20$ .

## 5.3. Ciclos: Hamiltoniano y Euleriano

Los ciclos (circuitos) en gráficos son muy interesantes para los aficionados a la IA, ya que están relacionados con problemas combinatorios intrigantes, algunos de ellos NP-Difíciles. En este sentido, parece conveniente explorar uno de estos problemas, el **Problema del Viajante (TSP)**, donde el estudio de dos tipos principales de ciclos, **los ciclos hamiltonianos y los ciclos eulerianos**, juegan un papel clave.

### 5.3.1. El problema del viajante

Un viajante de comercio optimiza sus visitas de la siguiente manera: Tiene que visitar  $n$  ciudades  $c_1, c_2, \dots, c_n$  **una vez** de tal manera que después de visitar la última ciudad el viajero regrese a la de partida (es decir, tenemos un ciclo). Este tipo de ciclo se llama **ciclo hamiltoniano**.

En la **versión métrica** del problema, cada ciudad tiene asociada una ubicación en el avión  $(x_i, y_i)$ , lo que significa que podemos explotar las distancias euclidianas entre ciudades para decidir la estrategia óptima. En realidad, el algoritmo debe encontrar el **ciclo de costo mínimo**. Un poco más formalmente, debemos encontrar una permutación  $\pi(1, \dots, n)$  de las ciudades para que

$$\pi = \arg \min d(\pi(1), \pi(2)) + d(\pi(2), \pi(3)) + \dots + d(\pi(n-1), \pi(n)) + d(\pi(n), \pi(1)).$$

dónde  $d(\pi(i), \pi(j)) = \sqrt{(x_{\pi(i)} - x_{\pi(j)})^2 + (y_{\pi(i)} - y_{\pi(j)})^2}$ . Desde que tenemos  $n!$  permutaciones de  $n$  ciudades, el hecho de buscar una permutación específica, la que conduzca al coste mínimo, hace que el problema sea NP-Difícil. Todo lo que podemos hacer es proponer **aproximaciones polinómicas razonables**.

A continuación, vamos a especificar los pasos del algoritmo de Christofides (ver [Christofides](#)). que es la aproximación

[Saltar al contenido principal](#)

### 🔔 Algoritmo 5.5 (Christophides)

**Entradas** Dado un gráfico **completo**  $G = (V, E, w)$  con contrapesos en los bordes (distancias)  $w : E \rightarrow \mathbb{R}$ , es decir  $w(i, j) = d(i, j)$ .

**Salida** Devuelve un ciclo hamiltoniano  $\Gamma$ .

1. Encuentre un **árbol de expansión mínima**  $T$ .
2. Dejar  $O$  ser el conjunto de nodos con grado impar en  $T$ . Encuentre una **combinación perfecta de costo mínimo**  $M$ .
3. Suma el conjunto de aristas de  $M \cup T$ .
4. Encuentra un **tour euleriano**  $\Gamma^0$ .
5. **Atajo**  $\Gamma^0$  hacer un ciclo hamiltoniano  $\Gamma$ .
6. **devolver**  $\Gamma$ .

Este algoritmo es el utilizado para aproximar el TSP en [la Fig. 5.18](#), donde la ruta del viajero es:

$$\Gamma = 0 \rightarrow 6 \rightarrow 1 \rightarrow 4 \rightarrow 8 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 2 \rightarrow 9 \rightarrow 0 .$$

En aras de la simplicidad, solo mostramos en la figura los bordes más cercanos a cada uno de los  $|V| = 10$  nodos en  $G$ .

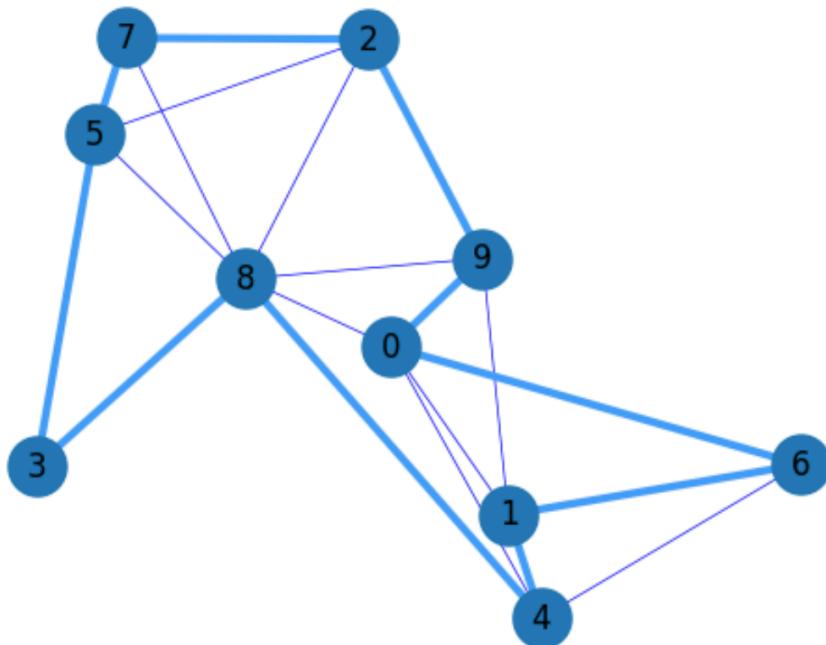


Fig. 5.18 Gráfico completo y solución TSP (en negrita).

#### 5.3.1.1. Árboles de expansión mínima

El primer paso del algoritmo TSP anterior consiste en calcular un **árbol de expansión mínima** (MST).

En primer lugar, un **árbol** no es más que *un gráfico sin ciclos*. Dados los nodos  $V$  de  $G = (V, E, w)$ , debemos encontrar un **subgrafo**  $T = (V, E', w)$ , con  $E' \subseteq E$ , satisfactorio:

- $T$  es un grafo conexo, es decir, **vincula todos los nodos** en  $V$ .
- $T$  es un **árbol**, es decir, no tiene ciclos/circuitos.
- La suma de pesos  $\sum_{e' \in E'} w(e')$  es **mínimo**.

Encontrar un MST (un árbol que abarque todos los nodos con un costo total mínimo) se puede hacer en  $O(|E| \log |E|)$ , o en  $O(|E| \log |V|)$  si  $G$  no tiene vértices aislados) de la siguiente manera (**Algoritmo de Kruskal**):

1. **Ordene** todos los bordes en orden ascendente según  $w(e)$ ,  $e \in E$ . Ponlos en una lista  $L$ .
2. Colocar  $T = \emptyset$ .

[Saltar al contenido principal](#)

- Seleccione el siguiente borde de  $L$ .
- Si no forma un bucle, inclúyelo en  $T$ .

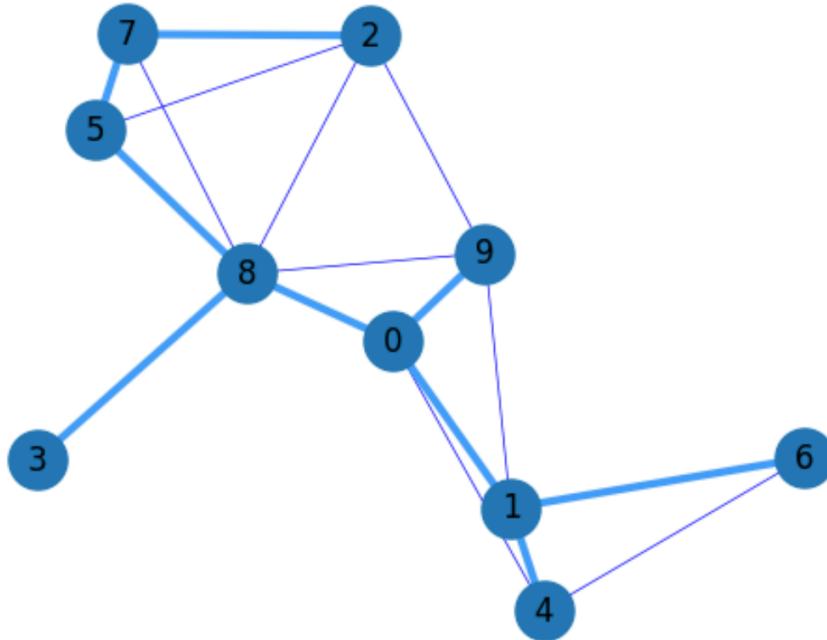


Fig. 5.19 TSP Fase 1: Calcular un MST.

Con respecto al problema de TSP, el propósito de encontrar un MST es en realidad encontrar una forma inicial de conectar todos los nodos en  $G$  con un costo mínimo. Por supuesto, esto no es un ciclo como podemos ver en [TSP2](#), ya que el TSP es acíclico, pero el algoritmo de Christofides extiende el TSP para encontrar un ciclo hamiltoniano en los pasos a seguir.

### 5.3.1.2. Combinaciones perfectas de peso mínimo

Dado un gráfico no dirigido  $G = (V, E)$ , una **coincidencia**  $M$  es un conjunto de aristas **no adyacentes**  $E' \subseteq E$  que no forman bucles. Como resultado:

1. los bordes en  $M$  **no comparten nodos comunes**.
2. el emparejamiento  $M$  se llama **máximo** si no está incluido en ninguna otra comparación.
3. el emparejamiento  $M$  se llama **perfecto** si une todos los nodos en  $G$ .
4. La **combinación perfecta de peso mínimo** es la combinación perfecta con un peso total mínimo.  $\sum_{e' \in M} w(e')$ . Este problema se puede resolver mediante el dual del **algoritmo Blossom de Edmond** con complejidad  $O(|V|^3)$  reemplazando primero los pesos de la siguiente manera  $w'(e) = (\max\{w(e) : e \in E\} + 1) - w(e)$  y luego encontrar la **combinación perfecta de peso máximo** con el algoritmo de Blossom.

Con respecto al problema MST, explotamos las siguientes propiedades de coincidencias perfectas:

1. Un gráfico sólo puede contener una coincidencia perfecta cuando el gráfico tiene un número **par** de nodos .
2. Siempre son posibles coincidencias perfectas en **gráficos completos** como el que utiliza el TSP. En realidad, el número de coincidencias perfectas en los gráficos completos (por supuesto con un número par de nodos) es el doble factorial del número par de nodos:  $|V|!!$ .

Por lo tanto, [antes de calcular la coincidencia perfecta de peso mínimo, eliminamos de  \$G\$  todos los nodos que tienen un grado impar en el MST  \$T\$](#) . Haciendo esto obtenemos aristas con los pesos más pequeños posibles. Como mostramos en [TSP3](#), donde agregamos la combinación perfecta de peso mínimo  $M$ , ser

$$M = \{(3,8), (4,1), (9,2), (0,6)\}$$

Obtenemos **incidentes de bordes en el MST**.

[Saltar al contenido principal](#)

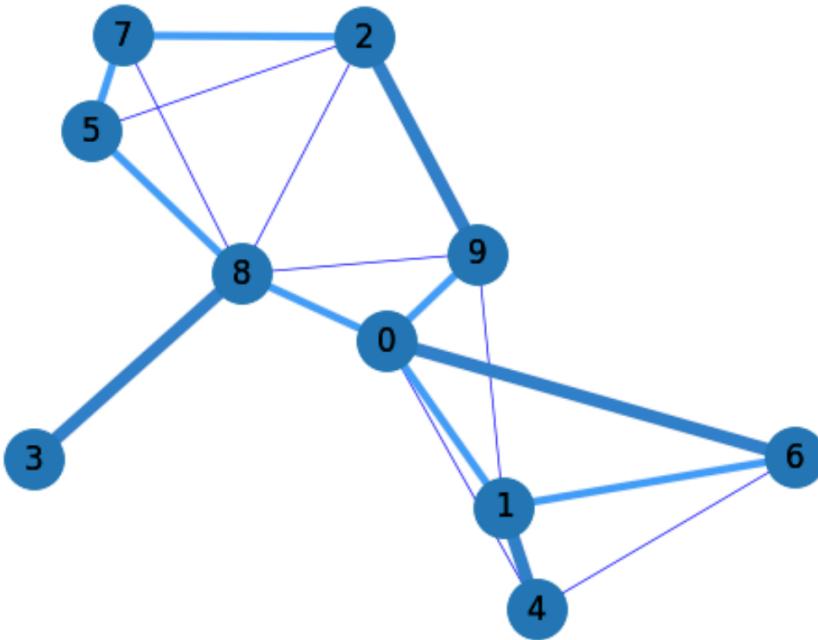


Fig. 5.20 TSP Fase 2: Agregue Perfect Matching de peso mínimo a MST.

### 5.3.1.3. Ciclos Eulerianos

Los ciclos eulerianos (circuitos o recorridos) son ciclos en los que cada vértice del gráfico se visita una vez. Cuando la gráfica no está dirigida, consideramos  $(u, v) \neq (v, u)$ . Por lo tanto, en el ejemplo anterior, podemos usar  $(3, 8)$  y  $(8, 3)$ , pero ninguno de ellos dos veces!

El nombre "euleriano" proviene del matemático Leonhard Euler, quien motivó el problema a través del famoso problema de los puentes de Konigsberg. Derivó la siguiente *condición necesaria* \* para la existencia de recorridos eulerianos en un grafo:

**Teorema de Euler** . Un gráfico conexo tiene un ciclo de Euler si y sólo si cada vértice tiene grado par .

Por supuesto, esta condición la cumple el gráfico TSP después de eliminar los vértices con grado impar. Entonces, se puede encontrar un recorrido euleriano con complejidad  $O(|E|^2)$  con el algoritmo de Fleury.

En particular, el recorrido euleriano encontrado por Networkx en el ejemplo anterior es:

$$\Gamma^0 = 0 \rightarrow 6 \rightarrow 1 \rightarrow 4 \rightarrow 1 \rightarrow 0 \rightarrow 8 \rightarrow 3 \rightarrow 8 \rightarrow 5 \rightarrow 7 \rightarrow 2 \rightarrow 9 \rightarrow 0 .$$

Todos los bordes en  $\Gamma^0$  pertenece a  $MST \cup M$ . Todo lo que tenemos que hacer para encontrar un recorrido hamiltoniano es eliminar **los nodos repetidos** (shortcuts) en  $\Gamma^0$ , resultando así la solución al MST:

$$\Gamma = 0 \rightarrow 6 \rightarrow 1 \rightarrow 4 \rightarrow 8 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 2 \rightarrow 9 \rightarrow 0 .$$

### 5.3.1.4. Optimidad de la aproximación

El algoritmo de Christofides devuelve una aproximación cuyo costo está acotado superiormente por  $3/2$  veces mayor que la de la solución óptima (que requiere un algoritmo no polinomial).

# Clasificación y particiones

## Contenido

- 6.1. Matriz de transición
- 6.2. Matrices laplacianas

### 6.1. Matriz de transición

Dado un gráfico (no dirigido)  $G = (V, E)$ , consideramos:

- La **matriz de adyacencia**  $\mathbf{A} = [a_{ij}] \in \{0, 1\}^{n \times n}$  de dimensión  $n \times n$ , donde  $n = |V|$  es el número de nodos, y

$$a_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- La **matriz de grados**  $\mathbf{D} = [d_i]$  es una  $n \times n$  matriz diagonal donde  $d_i = \sum_j a_{ij}$  es el grado  $\deg(i)$  de nodo  $i$ .

Entonces, si el grafo no tiene nodos aislados (ninguno de ellos tiene grado cero) se genera la llamada **matriz de transición**.  $\mathbf{P} = [p_{ij}]$ , que codifica la cadena de Markov asociada con los nodos del gráfico, viene dada por

$$\mathbf{P} = \mathbf{D}^{-1} \mathbf{A} \text{ i.e. } p_{ij} = \frac{a_{ij}}{d_i}.$$

Es muy interesante destacar las siguientes propiedades de  $\mathbf{P}$ :

- **Simplemente Estocástico (por filas)**. todas las filas  $i = 1, 2, \dots, n$  satisfacen:

$$\sum_{j=1}^n p_{ij} = 1.$$

es decir, cada fila  $\mathbf{P}_{i:}$  (asociado a un nodo  $i$ ) *puede interpretarse como una distribución de probabilidad*.

- **Rango deficiente**. En general,  $\mathbf{P}$  no es invertible, es decir  $\text{rank}(\mathbf{P}) \ll n$ , y cuanto mayor sea el rango, más diferentes (y por lo tanto informativas) serán las distribuciones de probabilidad de las filas.

#### 6.1.1. Distribución estacionaria

Cuando estudiamos las cadenas de Markov en el contexto de paseos aleatorios, dejamos de lado el uso de  $\mathbf{P}$  para caracterizar la cadena de Markov **en el límite**. Esto significa que podemos obtener información muy interesante sobre los nodos en  $G = (V, E)$  analizando **los poderes**  $\mathbf{P}^k$  para valores crecientes de  $k = 1, 2, \dots, \infty$ .

Empecemos por entender el significado de  $\mathbf{P}, \mathbf{P}^2, \mathbf{P}^3$ , etc. En realidad  $\mathbf{P}_{ij}^k$  es la probabilidad de llegar al nodo  $j$  de  $i$  en  $k$  pasos o **saltos**.

Dependiendo del gráfico  $G = (V, E)$  es posible que necesitemos solo una pequeña cantidad de poderes  $k$ . Considere, por ejemplo, el **gráfico completo** o el **gráfico de camarillas** para  $n$  nodos, denominados  $K_n$ . La gráfica completa es la **gráfica más densa que uno pueda imaginar** ya que tienen  $n$  nodos y cada nodo tiene  $n-1$  enlaces (incluidos los bucles automáticos). Por lo tanto, la matriz de adyacencia de  $K_n$  es:

$$\mathbf{A} = \mathbf{1}\mathbf{1}^T,$$

dónde  $\mathbf{1}$  es el  $n \times 1$  vector de todos unos, es decir  $\mathbf{A}$  es una matriz de unos.

Bueno, el grado de cada nodo en  $K_n$  es constante para cada nodo e igual a  $n$ . Por lo tanto, la matriz de transición es

[Saltar al contenido principal](#)

$$\mathbf{P} = \frac{1}{n} \mathbf{A}, \text{ with } p_{ij} = \frac{1}{n} \forall i, j .$$

Entonces cada nodo  $i$  tiene la misma probabilidad de ir a cualquier otro  $j$  (incluido él mismo) en un solo salto. En otras palabras, un nodo  $i$  no es necesario visitar primero ningún otro nodo antes de acceder a  $j$ . Esto hace  $\mathbf{P}^2, \mathbf{P}^3$ , etc, redundante desde

$$\mathbf{P}^2 = \frac{1}{n} \mathbf{1} \mathbf{1}^T \cdot \frac{1}{n} \mathbf{1} \mathbf{1}^T = \frac{1}{n^2} n \mathbf{1} \mathbf{1}^T = \frac{1}{n} \mathbf{1} \mathbf{1}^T = \mathbf{P} .$$

Esto significa que en  $K_n$  (así como en gráficos hiperdensos), todos los nodos tienen el mismo papel estructural. Tenga en cuenta que en estos casos tenemos  $\text{rank}(\mathbf{P}) = 1$ .

Sin embargo, en gráficos más generales podemos identificar o clasificar el papel estructural de cada nodo analizando  $\mathbf{P}^k$  para valores crecientes de  $k$ .

Considere, por ejemplo, un gráfico del modelo de bloques estocásticos (SBM) con  $n = 20$  nodos con probabilidad intraclasa  $p = 0.75$  y probabilidad entre clases  $q = 0.05$  (ver figura 6.1).

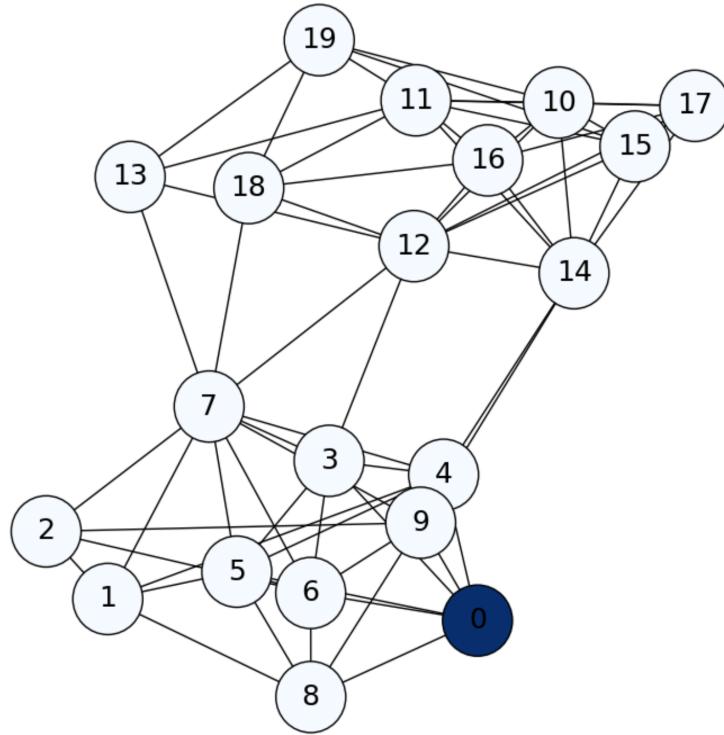


Fig. 6.1 Gráfico SBM. Difusión de  $i = 0$  (en la imagen) en  $\mathbf{f}^0$ .

Imaginemos que inyectamos información en el nodo  $i = 0$  y vea cómo dicha información se difunde a través del gráfico. Lo hacemos estableciendo un vector de probabilidad de fila,  $\mathbf{f}$ , como sigue:

$$\mathbf{f}^0 = [1 \ 0 \ \dots \ 0] .$$

Luego, descomponiéndose  $\mathbf{P} = [\mathbf{P}_{:,1} \ \mathbf{P}_{:,2} \ \dots \ \mathbf{P}_{:,n}]$  en términos de sus columnas  $\mathbf{P}_{:,j}$  tenemos

$$\mathbf{f}^1 = \mathbf{f}^0 \mathbf{P} = [\mathbf{f}^0 \mathbf{P}_{:,1} \ \mathbf{f}^0 \mathbf{P}_{:,2} \ \dots \ \mathbf{f}^0 \mathbf{P}_{:,n}]$$

donde, comenzando en un nodo general il lleva a

$$\mathbf{f}_i^1 = \sum_{j=1}^n \mathbf{f}_i^0 p_{ij} = \sum_{j \in \mathcal{N}_i} \mathbf{f}_i^0 p_{ij} .$$

En particular, para  $i = 1$  tenemos un vector de fila  $\mathbf{f}^1$  con  $f_i^1 = p_{1j}$ , es decir, podemos llegar a cualquiera de los otros nodos  $j$ .

[Saltar al contenido principal](#)

nodo $i = 1$ .

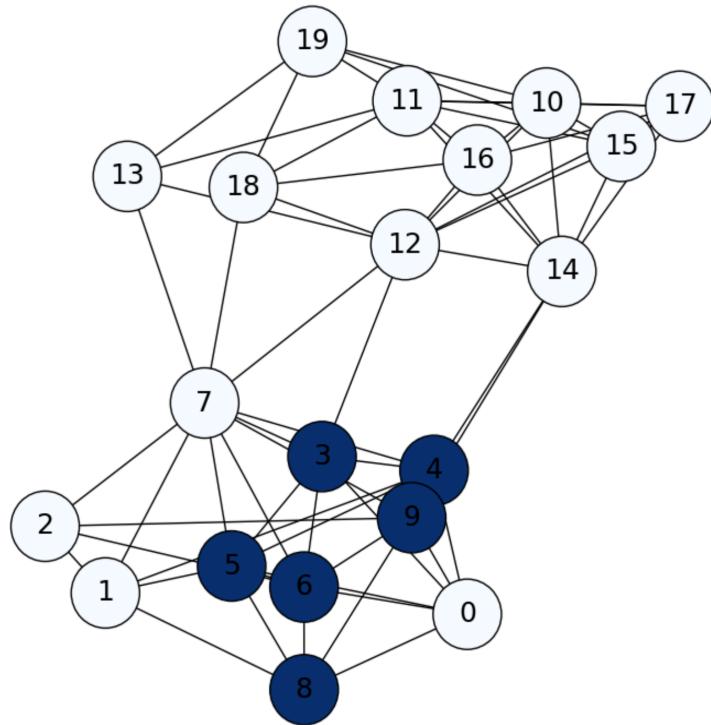


Fig. 6.2 Gráfico SBM. Difusión de  $i = 1$  (en la imagen) en  $\mathbf{f}^1$ .

Repitamos de nuevo:

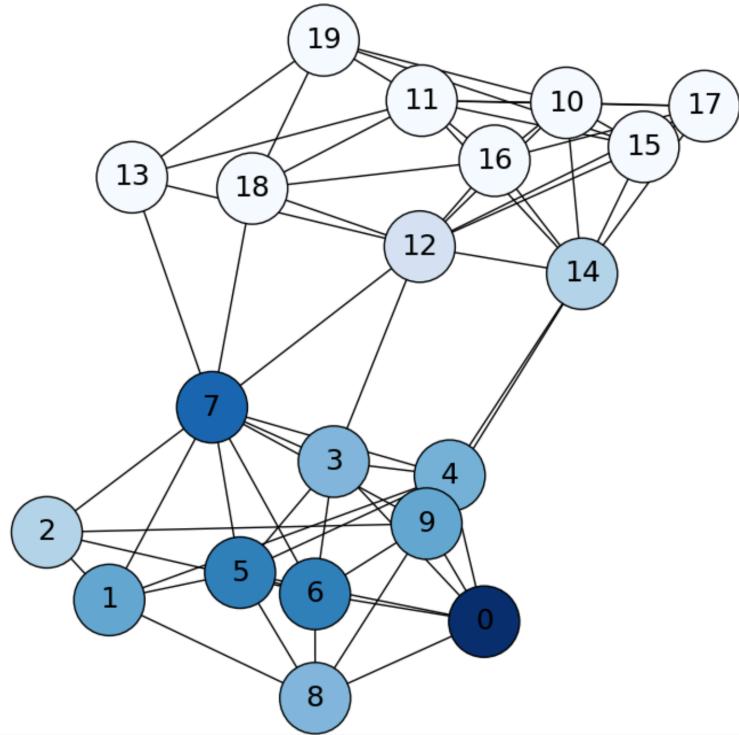
$$\mathbf{f}_i^2 = \sum_{j=1}^n \mathbf{f}_i^1 p_{ij} = \sum_{j \in \mathcal{N}_i} \mathbf{f}_i^1 p_{ij} .$$

Desde,

$$\mathbf{f}^1 \mathbf{P} = (\mathbf{f}^0 \mathbf{P}) \mathbf{P} = \mathbf{f}^0 \mathbf{P}^2 ,$$

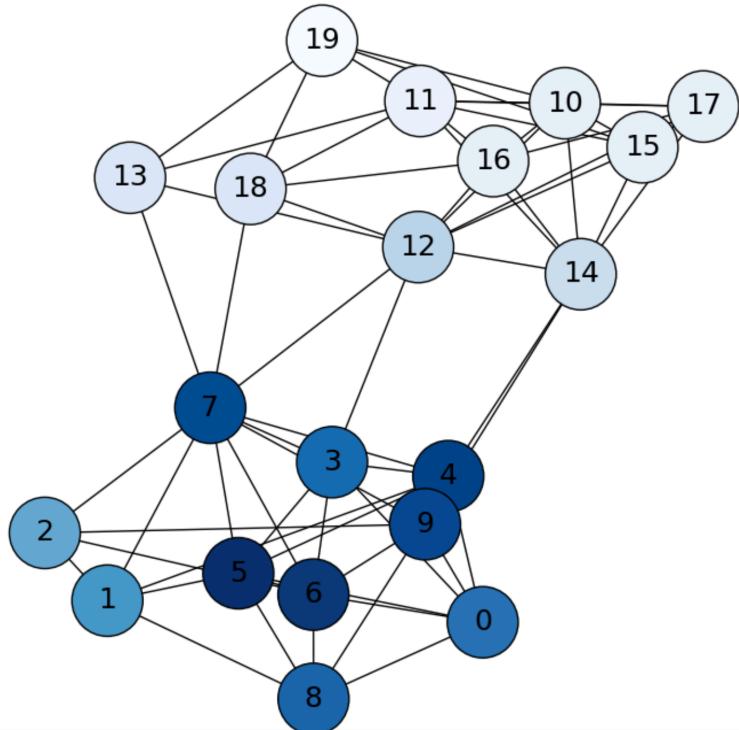
podemos llegar a los **vecinos de segundo orden** del nodo $i$ , es decir, los nodos que están a **dos saltos de** $i$ , en particular algunos nodos de otra comunidad, como vemos en [la Fig. 6.3](#).

[Saltar al contenido principal](#)

Fig. 6.3 Gráfico SBM. Difusión de  $i = 1$  (en la imagen) en  $\mathbf{f}^2$ .

Iterando nuevamente afirmamos que es más probable que estemos en la comunidad de  $i = 1$ , pero podemos saltar a otra comunidad (ver [Fig. 6.4](#)):

$$\mathbf{f}_i^3 = \sum_{j=1}^n \mathbf{f}_i^2 p_{ij} = \sum_{j \in \mathcal{N}_i} \mathbf{f}_i^2 p_{ij} .$$

Fig. 6.4 Gráfico SBM. Difusión de  $i = 1$  (en la imagen) en  $\mathbf{f}^3$ .

Entonces surgen dos preguntas interesantes:

[Saltar al contenido principal](#)

## 2. ¿Es esa probabilidad estabilizada la misma para todas las opciones de nodo inicial? ?

**Distribución estacionaria**. Bajo ciertas condiciones (ver más abajo) la respuesta a ambas preguntas es **Sí**.

Para  $t \rightarrow \infty$  tenemos eso si alcanzamos una distribución de filas determinada  $\pi$ , la resultante  $\pi \mathbf{P}$  es exactamente  $\pi$ , donde

$$\pi = \pi \mathbf{P} \text{ and } \pi_i = \frac{d_i}{\text{vol}(G)},$$

$\text{vol}(G) = \sum_{i=1}^n d_i$  es el **volumen** de la gráfica ( $\text{vol}(G) = 2|E|$  para gráficos no dirigidos). En otras palabras,  $\pi$  es un **punto fijo** conocido como **distribución estacionaria** de  $\mathbf{P}$ . En realidad es muy sencillo demostrar que

$$d_i = \sum_{j \in \mathcal{N}_i} d_i p_{ij} = \sum_{j \in \mathcal{N}_i} \frac{d_i}{d_i} = \sum_{j \in \mathcal{N}_i} 1 = d_i.$$

Entonces, la distribución estacionaria  $\pi$  es el resultado de normalizar la distribución de grados por el volumen.

En nuestro ejemplo, mostramos la distribución estacionaria de nuestro SBM en [la Fig. 6.5](#).

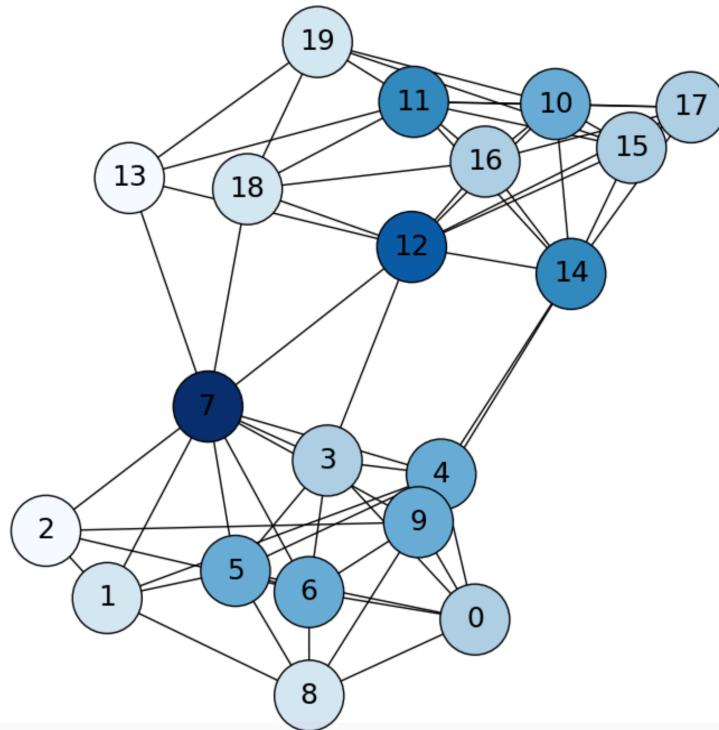


Fig. 6.5 Gráfico SBM. Distribución estacionaria  $\pi$ .

En otras palabras, alcanzar la distribución estacionaria en saltos significa que desde cualquier **tiempo discreto**  $t + \Delta t$ , la **probabilidad de estar en cualquier nodo  $i$**  sólo depende de su grado  $d_i$ .

Para aclarar este punto, tenga en cuenta que tenemos que

$$\lim_{t \rightarrow \infty} \mathbf{P}^t = \Pi$$

y  $\Pi$  es una matriz donde cada fila es una copia de la distribución estacionaria  $\pi$ . Mire por ejemplo la fila  $i$ :

$$\Pi_{i:} = \pi = \left[ \frac{d_1}{\text{vol}(G)} \frac{d_2}{\text{vol}(G)} \cdots \frac{d_n}{\text{vol}(G)} \right],$$

es decir, la probabilidad de pasar de  $i$  a cualquier otro nodo  $j$  es proporcional a  $d_j$ , **independientemente de si** es vecino de  $i$  o no!

[Saltar al contenido principal](#)

Consideremos ahora el gráfico completo  $K_n$ , donde cada nodo puede visitar cualquier otro (incluido él mismo) con probabilidad  $\frac{1}{n} = \frac{n}{n^2} = \frac{d_i}{\sum_i d_i}$ . En otras palabras, una gráfica completa es una gráfica que ha alcanzado su distribución estacionaria en el tiempo  $t = 0$ .

**Condiciones formales**. La distribución estacionaria es única si:

1. La cadena de Markov es **irreductible** (todos los estados son mutuamente accesibles). Para un gráfico no dirigido es suficiente que el gráfico sea conexo. Para los dígrafos debemos asegurarnos de que exista una ruta dirigida entre cualquier par de nodos, lo cual es más difícil de lograr en redes realistas.
2. La cadena de Markov es **aperiodica**. Para satisfacer esta condición, necesitamos que si existe un número entero positivo  $k$  st todas las entradas en  $\mathbf{P}^k$  son mayores que cero. Una condición menos estricta es tener al menos un nodo con un bucle propio.

## 6.1.2. La interpretación espectral: PageRank

La distribución estacionaria  $\pi$ , remodelado como un  $n \times 1$  El vector puede considerarse como un **vector propio izquierdo** de la matriz de transición para el **valor propio**.  $\lambda = 1$  desde

$$\phi^T \mathbf{P} = \lambda \phi^T \text{ leads to } \pi^T \mathbf{P} = 1 \cdot \pi^T .$$

De manera similar, los vectores propios y **valores propios correctos** (normalmente los calculamos) se definen como:

$$\mathbf{P}\psi = \lambda\psi \text{ leads to } \mathbf{P}\psi = 1 \cdot \psi .$$

Observando las ecuaciones anteriores, en el siguiente tema demostraremos que  $\psi = \mathbf{D}^{1/2}\mathbf{1}$  es el vector propio derecho de la matriz de transición  $\mathbf{P}$ .

El significado de  $\psi = \mathbf{D}^{1/2}\mathbf{1}$  es eso

$$\psi_i = \sqrt{d_i}$$

está relacionado con  $\pi_i = \frac{d_i}{\text{vol}(G)}$ , en términos de que es una función monótona de  $d_i$ . Ambos  $\pi$  y  $\psi$  **Clasifique** la importancia de un nodo en el gráfico en términos de su grado.

Como veremos a continuación, calcular el vector propio correcto es muy interesante en entornos prácticos, ya que **puede aproximarse incluso en redes/gráficos que no satisfacen las condiciones formales de irreductibilidad y aperiodicidad**.

**Teorema de Perron-Frobenius**. Si  $M$  es una matriz estocástica de columna positiva , entonces:

1.  $\lambda = 1$  es un valor propio de multiplicidad uno (único).
2.  $\lambda = 1$  es el valor propio más grande: todos los demás valores propios tienen un valor absoluto menor que 1.
3. Los vectores propios correspondientes al valor propio  $\lambda = 1$  tener solo entradas positivas o solo entradas negativas.
4. En particular, para el valor propio  $\lambda = 1$  existe un vector propio único con la suma de sus entradas igual a 1 (una distribución de probabilidad).

Entonces  $M = P^T$  satisface este teorema. Como en general toda matriz cuadrada tiene los mismos valores propios que su transpuesta, aproximaremos el vector propio izquierdo de  $P$  por computación

$$\mathbf{P}^T \phi = (\phi^T \mathbf{P})^T = 1 \cdot \phi .$$

y en la práctica podemos aproximarnos al vector propio  $\phi$  asociado a  $\lambda = 1$  de forma iterativa. Para ello utilizamos el **Método Power** . Como mostramos en Power , todo lo que hacemos es:

**Algoritmo 6.1 (Método de potencia)**

**Entradas** Dada una  $n \times n$  matriz  $M$  y  $\mathbf{x} = \frac{1}{n}\mathbf{1}$

**Salida** Calcular una aproximación del vector propio principal.  $\phi$

1. **para** cada uno de los  $k$  iteraciones:

1.  $\mathbf{x} \leftarrow M\mathbf{x}$
2. Normalizar:  $\mathbf{x} \leftarrow \frac{\mathbf{x}}{\|\mathbf{x}\|}$

2. **devolver**  $\mathbf{x}$

1. Inicializar  $\mathbf{x}_i = \frac{1}{n}$  (estimación completamente aleatoria del vector propio).

2. Para  $M = P^T$  calcular  $M\mathbf{x}, M^2\mathbf{x}, \dots, M^k\mathbf{x}$ .

3. Entre una potencia y la siguiente, normalizar el vector propio actual  $\mathbf{x}$ .

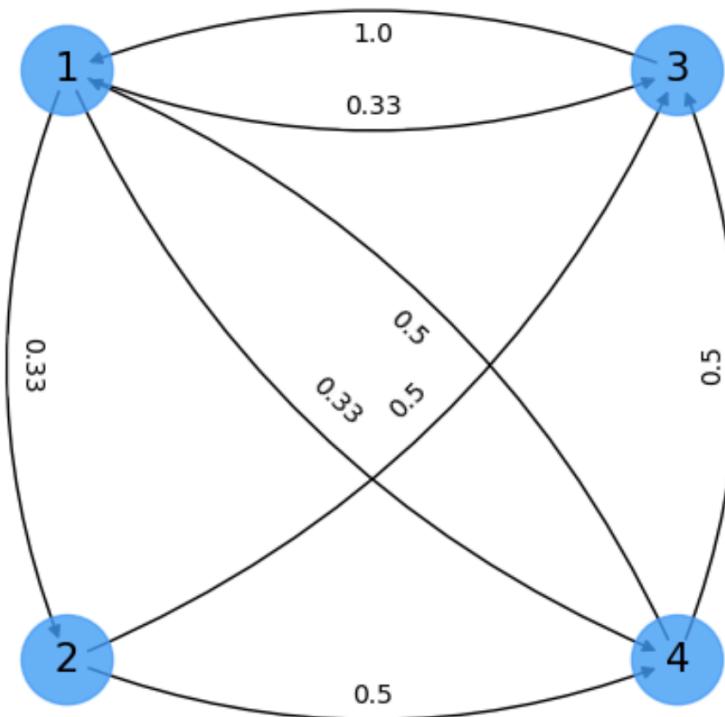


Fig. 6.6 Pequeño dígrafo para calcular el vector propio principal.

**Ejercicio**. Nuestro dígrafo Fig. 6.6 no es aperiódico. Este problema no es tan difícil como el de no ser irreducible. En particular, todos los nodos son mutuamente accesibles y esto nos permite obtener una buena aproximación.

$$\mathbf{x}_0 = \frac{1}{n}\mathbf{1} = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} \quad \mathbf{x}_1 = P^T \mathbf{x}_0 = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} = \begin{bmatrix} 0.375 \\ 0.083 \\ 0.333 \\ 0.208 \end{bmatrix}$$

$$\mathbf{x}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} = \begin{bmatrix} 0.682 \\ 0.151 \\ 0.606 \\ 0.379 \end{bmatrix} \quad \mathbf{x}_2 = P^T \mathbf{x}_1 = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.682 \\ 0.151 \\ 0.606 \\ 0.379 \end{bmatrix} = \begin{bmatrix} 0.796 \\ 0.227 \\ 0.492 \\ 0.303 \end{bmatrix}$$

[Saltar al contenido principal](#)

$$\mathbf{x}_2 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} = \begin{bmatrix} 0.788 \\ 0.225 \\ 0.487 \\ 0.300 \end{bmatrix} \quad \mathbf{x}_2 = \mathbf{P}^T \mathbf{x}_1 = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.788 \\ 0.225 \\ 0.487 \\ 0.300 \end{bmatrix} = \begin{bmatrix} 0.637 \\ 0.262 \\ 0.525 \\ 0.375 \end{bmatrix}$$

$$\text{en iteración 10 tendremos } \mathbf{x}_9 = \frac{\mathbf{x}_8}{\|\mathbf{x}_8\|} = \begin{bmatrix} 0.720 \\ 0.240 \\ 0.541 \\ 0.361 \end{bmatrix} \quad \mathbf{x}_{10} = \mathbf{P}^T \mathbf{x}_9 = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.720 \\ 0.240 \\ 0.541 \\ 0.361 \end{bmatrix} = \begin{bmatrix} 0.722 \\ 0.240 \\ 0.540 \\ 0.360 \end{bmatrix}.$$

Por lo tanto, el nodo más importante es el nodo 1, seguido de nodos 3, 4 y 2. Tenga en cuenta que este resultado es bastante diferente de la distribución estacionaria estimada en términos de **grados**:

$$\pi = \frac{\mathbf{D}_{out}\mathbf{1}}{\text{vol}(G)} = \begin{bmatrix} 0.375 \\ 0.25 \\ 0.125 \\ 0.25 \end{bmatrix}.$$

Y también diferente del vector propio derecho teórico:

$$\psi = \sqrt{\mathbf{D}_{out}}\mathbf{1} = \begin{bmatrix} 1.732 \\ 1.414 \\ 1.000 \\ 1.414 \end{bmatrix}.$$

Aunque en ambos casos se detecta correctamente el nodo mejor clasificado.

**El PageRank de Google**. Una de las aplicaciones más representativas del método del poder es encontrar el ranking de páginas web en Internet. Cuanto mayores sean los valores (en valor absoluto) de los componentes del vector propio principal resultante **x** cuanto más importante es la página .

Sin embargo, Internet (que puede considerarse un grafo gigantesco cuyos nodos son las páginas web), no necesariamente reúne las condiciones formales necesarias para la unicidad o convergencia de **x**.

Por ejemplo, en [la Fig. 6.7](#) resumimos los principales problemas prácticos que surgen en redes realistas:

- **Fuentes**: nodos solo sin grado, como nodo 2 que sólo está vinculado a 3 y 4.
- **Sumideros**: nodo con grado de salida cero, como el nodo 3.

Además podemos tener **nodos aislados** o gráficos fragmentados en muchos **componentes conectados**.

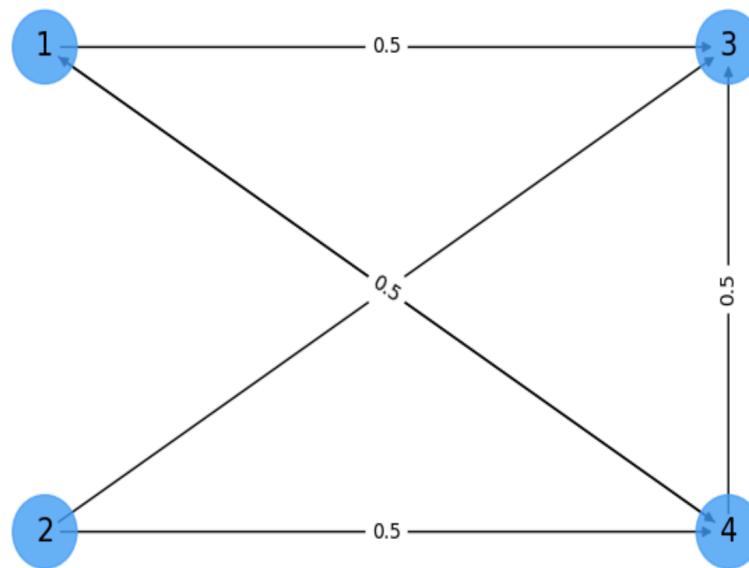


Fig. 6.7 Rango de páginas. Necesidad de la matriz de Google.

La solución a estos problemas encontrada por Page y Brin al diseñar un modelo para el internauta de Google, fue complementar la matriz de transición (en realidad su transpuesta), con un gráfico completo de modo que con cierta probabilidad  $1 - \alpha$  (típicamente  $1 - \alpha = 0.85$ ) seguimos la matriz de transición, pero con una pequeña probabilidad  $\alpha = 0.15$  (denominada **probabilidad de teletransportación**) saltamos aleatoriamente a cualquier otro nodo o permanecemos en el mismo modo (autobucle). Por tanto, la llamada **matriz de Google**:

$$\mathbf{G} = (1 - \alpha) \cdot \mathbf{P}^T + \alpha \cdot \frac{1}{n} \mathbf{1}^T \mathbf{1}$$

Por tanto, la matriz de Google consiste en añadir **arestas virtuales** entre cualquier par de nodos (incluidos los autobuses). Esto hace que la cadena de Markov resultante sea irreductible y facilita el flujo de información. En este sentido, encontrar cuál es la probabilidad de que un navegante termine haciendo clic en una página determinada se realiza mediante el método Power.

**Ejercicio**. Repetimos el ejercicio anterior utilizando la matriz de Google para el dígrafo de la Fig. 6.7 .

$$\mathbf{x}_0 = \frac{1}{n} \mathbf{1} = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} \quad \mathbf{x}_1 = \mathbf{Gx}_0 = \begin{bmatrix} \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} & (1 - \alpha) \frac{1}{2} \\ \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} \\ (1 - \alpha) \frac{1}{2} & (1 - \alpha) \frac{1}{2} & \alpha \frac{1}{4} & (1 - \alpha) \frac{1}{2} \\ (1 - \alpha) \frac{1}{2} & (1 - \alpha) \frac{1}{2} & \alpha \frac{1}{4} & \alpha \frac{1}{4} \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} = \begin{bmatrix} \alpha \frac{1}{16} + \frac{2}{16} \\ \alpha \frac{4}{16} \\ \frac{6}{16} - \alpha \frac{5}{16} \\ \frac{4}{16} - \alpha \frac{2}{16} \end{bmatrix}$$

$$\mathbf{x}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} = \begin{bmatrix} 0.312 \\ 0.081 \\ 0.774 \\ 0.543 \end{bmatrix} \quad \mathbf{x}_2 = \mathbf{Gx}_1 = \begin{bmatrix} \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} & (1 - \alpha) \frac{1}{2} \\ \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} \\ (1 - \alpha) \frac{1}{2} & (1 - \alpha) \frac{1}{2} & \alpha \frac{1}{4} & (1 - \alpha) \frac{1}{2} \\ (1 - \alpha) \frac{1}{2} & (1 - \alpha) \frac{1}{2} & \alpha \frac{1}{4} & \alpha \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0.312 \\ 0.081 \\ 0.774 \\ 0.543 \end{bmatrix} = \begin{bmatrix} 0.295 \\ 0.064 \\ 0.462 \\ 0.231 \end{bmatrix}$$

en iteración 10 tendremos

$$\mathbf{x}_9 = \frac{\mathbf{x}_9}{\|\mathbf{x}_9\|} = \begin{bmatrix} 0.264 \\ 0.066 \\ 0.484 \\ 0.286 \end{bmatrix} \quad \mathbf{x}_{10} = \mathbf{Gx}_9 = \begin{bmatrix} \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} & (1 - \alpha) \frac{1}{2} \\ \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} & \alpha \frac{1}{4} \\ (1 - \alpha) \frac{1}{2} & (1 - \alpha) \frac{1}{2} & \alpha \frac{1}{4} & (1 - \alpha) \frac{1}{2} \\ (1 - \alpha) \frac{1}{2} & (1 - \alpha) \frac{1}{2} & \alpha \frac{1}{4} & \alpha \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0.264 \\ 0.066 \\ 0.484 \\ 0.286 \end{bmatrix} = \begin{bmatrix} .422 \\ .105 \\ .774 \\ .458 \end{bmatrix}$$

Ahora, el nodo mejor clasificado es 3, el que tiene mayor **grado de entrada**. Nodos 1 y 4 son menos accesibles. Curiosamente, el

[Saltar al contenido principal](#)

En resumen, una propiedad importante del algoritmo Pagerank es que al introducir el  $\alpha \frac{1}{n}$  términos (conocido como **teletransporte**) confiere a la matriz de Google las propiedades de **irreductibilidad y aperiodicidad**. El gráfico es **irreducible** ya que se puede llegar a todos los nodos del gráfico mediante recorridos aleatorios. Además es **aperiodico** porque se cierra con peso  $\alpha \frac{1}{n}$  se agregan si no existen; si existen, su peso es  $\frac{(1-\alpha)}{d_i} + \alpha \frac{1}{n}$ . De todos modos, como todos los nodos tienen bucles automáticos, su período es 1 y la gráfica es aperiódica.

## 6.2. Matrices laplacianas

Hasta ahora, hemos aprendido que los vectores propios con valores propios máximos explican o [resumen la distribución estacionaria de un gráfico](#) mediante el análisis de su matriz de transición (transpuesta).

Sin embargo, algunas otras propiedades de las gráficas se pueden entender mejor si analizamos los vectores propios y valores propios de otras matrices como la **laplaciana** o la **laplaciana normalizada**. En particular, estas matrices son clave para [descubrir propiedades combinatorias de gráficos](#).

Pero, ¿qué significa aquí **combinatoria**? Consideremos, por ejemplo, el mismo SBM estudiado anteriormente.  $G = (V, E)$  y **dos colores**, digamos azul y rojo, codificados respectivamente por los números  $-1$  y  $+1$  (codificación bipolar). Entonces surge la siguiente pregunta fundamental:

**Cómo colorear los nodos** de  $G = (V, E)$ ? Para que las diferencias de color aparezcan sólo en los **bordes más débiles**?

### 6.2.1. problema del corte mínimo

**Corte mínimo**. Colorear significa **dividir**: los nodos con el mismo color pertenecen a la *misma partición*. Si solo tenemos dos colores, entonces el conjunto de nodos es la unión de dos *subconjuntos disjuntos*, es decir

$$V = A \cup B, \text{ with } A \subseteq V, B \subseteq V : A \cap B = \emptyset$$

Por ejemplo, en [la figura 6.8](#), los nodos azules  $A$  (con etiqueta  $-1$ ) pertenecen a la comunidad inferior del SBM, mientras que los rojos  $B$  (con etiqueta  $+1$ ) pertenecen a la comunidad superior. Subconjuntos  $A$  y  $B$  inducen una *partición* ya que su unión es el conjunto completo de nodos y son disjuntos. En realidad, dicha partición es **óptima** ya que las diferencias de color aparecen en los **bordes más débiles**, aquellos [cuya eliminación desconecta o corta el gráfico maximizando la coherencia de las comunidades o clusters resultantes mientras se minimiza el número de bordes eliminados](#).

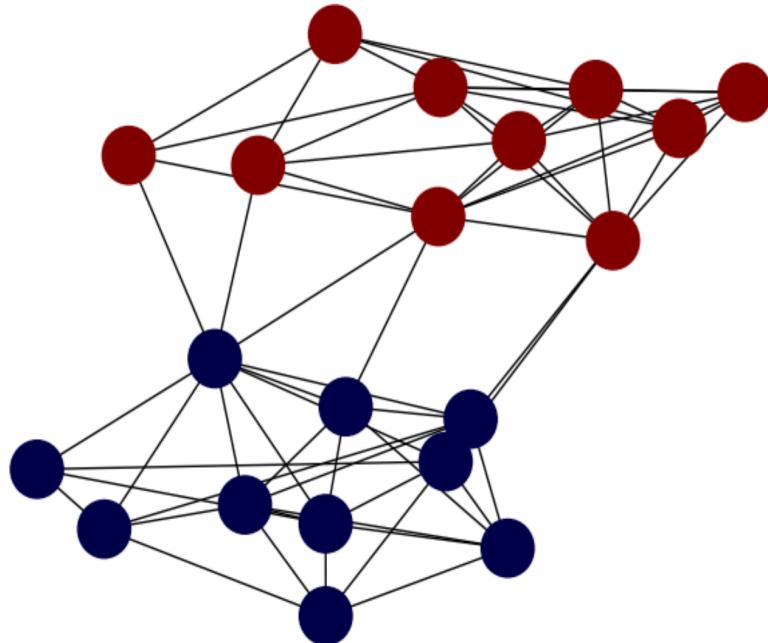


Fig. 6.8 Gráfico SBM. Partición óptima de corte mínimo.

Un poco más formalmente, definamos el  $\text{cut}(A, B)$  entre los dos subconjuntos  $A$  y  $B$  de una partición como el número de aristas con diferentes colores en sus nodos:

[Saltar al contenido principal](#)

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} e(u, v) \text{ with } e(u, v) \in E.$$

La idea de min-cut es encontrar la partición  $(A, B)$  **minimizando**  $\text{cut}(A, B)$ . Sin embargo, hacerlo no es suficiente porque si un nodo determinado  $u$  tiene  $d_u = 1$ , podemos considerar la siguiente partición:  $A = \{u\}$  y  $B = V - \{u\} = V - A$ , llevando a  $\text{cut}(A, B) = 1$  lo cual es mínimo pero sin sentido.

Entonces, parece más razonable forzar el equilibrio de la **partición**. Esto se puede hacer de forma sencilla **minimizando** el llamado **corte medio o corte ratio**.  $\text{Rcut}(A, B)$ :

$$\text{Rcut}(A, B) = \frac{\text{cut}(A, B)}{|A|} + \frac{\text{cut}(A, B)}{|B|},$$

dónde  $|A|$  y  $|B|$  son el número de nodos de  $A$  y  $B$  respectivamente, que se **maximizan** como  $\text{cut}(A, B)$  se minimiza.

Sin embargo, como hay  $2^{|V|}$  particiones o formas de colorear  $|V|$  nodos con 2 colores, el problema de **corte de proporción** es **NP-Duro y solo podemos tener aproximaciones**.

## 6.2.2. vector de Fiedler

**Codificación de particiones**. Dada una partición  $(A, B)$ , desde  $|A| + |B| = |V|$  podemos codificarlo con un vector  $\mathbf{f}$  de longitud  $|V|$  cuyas entradas  $f(u) \in \{-1, 1\}$  para  $u \in V$ . Por ejemplo, considere el gráfico de [la figura 6.9](#) y su partición:  $A = \{2, 6\}$ ,  $B = \{1, 3, 4, 5\}$ .

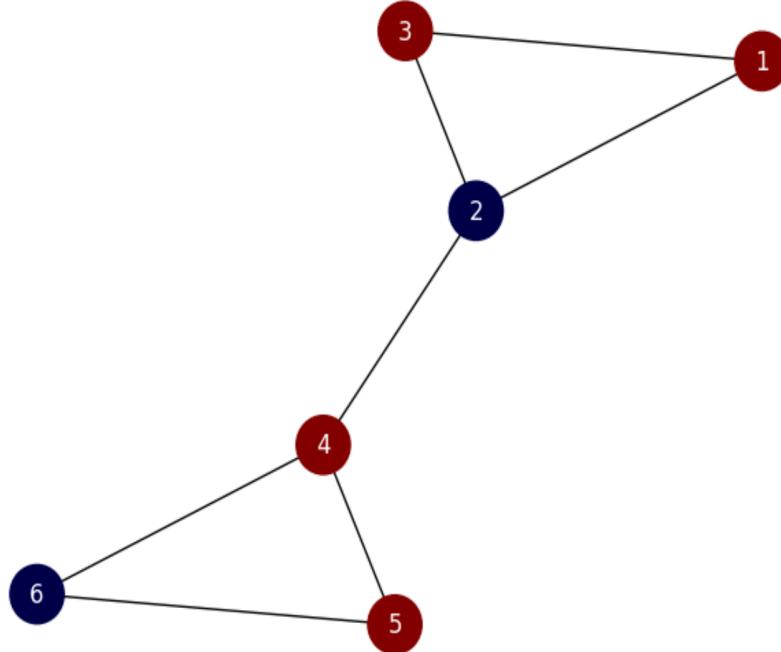


Fig. 6.9 Gráfico pequeño con partición no óptima.

El vector correspondiente es

$$\mathbf{f} = [+1 \ -1 \ +1 \ +1 \ +1 \ -1]^T.$$

Luego, para calcular la relación de corte de la partición anterior, contamos el número de **aristas heterófilas** (aristas que conectan nodos de diferentes colores). Ellos son  $E_{cut} = \{(1, 2), (2, 3), (2, 4), (5, 6)\}$ . Entonces,  $\text{cut}(A, B) = |E_{cut}| = 4$ . Entonces, desde  $|A| = 2$  y  $|B| = 4$  la relación de corte es

$$\text{Rcut}(A, B) = \frac{|E_{cut}|}{|A|} + \frac{|E_{cut}|}{|B|} = \frac{4}{2} + \frac{4}{4} = 3.$$

Considere ahora la partición que se muestra en [la figura 6.10](#).

[Saltar al contenido principal](#)

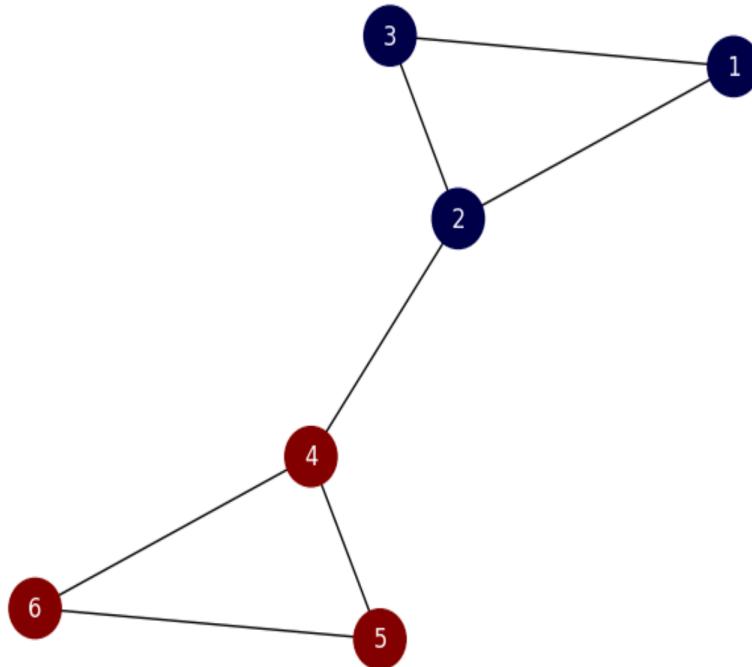


Fig. 6.10 Gráfico pequeño con la partición óptima.

tenemos el vector

$$\mathbf{f} = [-1 \ -1 \ -1 \ +1 \ +1 \ +1]^T$$

y

$$\text{Rcut}(A, B) = \frac{|E_{cut}|}{|A|} + \frac{|E_{cut}|}{|B|} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3},$$

que es claramente menor que la de la partición anterior:  $\frac{2}{3} < 3$ , lo que muestra que la **función de pérdida** de corte de proporción mide la bondad de un corte.

**Sin embargo, la función de pérdida** de ratio-cut es propensa a **soluciones degeneradas** como  $A = \{1, 2, 3, 4, 5, 6\}$ ,  $B = \emptyset$  o viceversa:

$$\text{Rcut}(A, B) = \frac{0}{|A|} + \frac{0}{|B|} = 0.$$

En consecuencia, cualquier algoritmo que minimice  $\text{Rcut}(A, B)$  debe:

1. encontrar un vector  $\mathbf{f} \in \{-1, 1\}^{|V|}$  codificando una partición  $(A, B)$  tal que **minimice**  $\text{Rcut}(A, B)$ .
2. El vector minimizador debe ser **perpendicular/ortogonal** a  $\mathbf{1}$  así como diferente de  $\mathbf{0}$ .

La última condición (ortogonalidad) asegura que la solución sea bastante diferente de la solución degenerada .

Por ejemplo,

$$\mathbf{f} = [-1 \ -1 \ -1 \ +1 \ +1 \ +1]^T$$

es ortogonal a  $\mathbf{1}$  desde el **producto escalar**  $\mathbf{f} \cdot \mathbf{1} = 0$ . Sin embargo,

$$\mathbf{f} = [+1 \ -1 \ +1 \ +1 \ +1 \ -1]^T.$$

no es ortogonal a  $\mathbf{1}$  desde  $\mathbf{f} \cdot \mathbf{1} = 4 - 2 = 2 \neq 0$ .

[Saltar al contenido principal](#)

**Relajación continua**. Debido a la dureza NP del problema de relación de corte, en lugar de buscar soluciones discretas,  $\mathbf{f} \in \{-1, 1\}^{|V|}$  debemos **relajar el espacio vectorial para permitir**  $\mathbf{f} \in [-1, 1]^{|V|}$ . Esta estrategia es bastante común en la IA y se llama **relajación continua**. La idea principal es que los vectores con entradas continuas son más fáciles de manipular mediante técnicas algebraicas para que se conviertan en una aproximación de sus contrapartes discretas (la relajación realmente significa "aproximación").

[Miroslav Fiedler](#), mientras razonaba sobre cómo utilizar el álgebra lineal para caracterizar la conectividad de gráficos, propuso la siguiente pérdida equivalente al ratio-cut:

$$\lambda = |V| \cdot \left\{ \frac{\sum_{u \sim v} (\mathbf{f}(u) - \mathbf{f}(v))^2}{\sum_{u,v} (\mathbf{f}(u) - \mathbf{f}(v))^2} : \mathbf{f} \neq \mathbf{0}, \mathbf{f} \neq c \cdot \mathbf{1}, c \in \mathbb{R} \right\},$$

donde la notación  $u \sim v$  significa que " $u$  y  $v$  somos vecinos". Como resultado, si  $\mathbf{f} \in \{-1, 1\}^{|V|}$ , el **numerador** de la pérdida anterior simplemente cuenta cuántas aristas heterófilas tenemos. El **denominador** cuenta cuántas aristas heterófilas tendríamos si el gráfico estuviera completo.

la correspondencia de  $\lambda$  con la relación de pérdida de corte es muy interesante. En realidad, todo lo que tenemos que hacer es operar con el denominador de la siguiente manera:

1. Utilice la **identidad de Lagrange**, que es válida para cualquier vector  $\mathbf{f}$ , para expandir el denominador

$$\frac{1}{2} \sum_{u,v} (\mathbf{f}(u) - \mathbf{f}(v))^2 = |V| \sum_u \mathbf{f}(u)^2 - \left( \sum_u \mathbf{f}(u) \right)^2.$$

2. Dado que la ortogonalidad de  $\mathbf{f} \cdot \mathbf{1}$  significa que  $\sum_u \mathbf{f}(u) = 0$ , la identidad de Lagrange se convierte en:

$$\sum_{u,v} (\mathbf{f}(u) - \mathbf{f}(v))^2 = 2|V| \sum_u \mathbf{f}(u)^2.$$

3. Mira ahora el corte de razones y agrupa las dos fracciones en una sola:

$$\begin{aligned} \text{Rcut}(A, B) &= \frac{|E_{cut}|}{|A|} + \frac{|E_{cut}|}{|B|} \\ &= \frac{|B| \cdot |E_{cut}| + |A| \cdot |E_{cut}|}{|A| \cdot |B|} \\ &= \frac{(|A| + |B|) \cdot |E_{cut}|}{|A| \cdot |B|} \\ &= \frac{|V| \cdot |E_{cut}|}{|A| \cdot |B|} \end{aligned}.$$

Ahora, estableciendo  $|E_{cut}| = \sum_{u \sim v} (\mathbf{f}(u) - \mathbf{f}(v))^2$  y considerando que  $\sum_u \mathbf{f}(u)^2 \leq |V|$  para  $\mathbf{f} \in [-1, 1]^{|V|}$ , tenemos eso

$$\lambda \leq \text{Rcut}(A, B)$$

esa es la pérdida  $\lambda$ ; Es **incluso más restrictivo que la proporción de** pérdida de corte!

**Ejercicio**. Para **comprender mejor la función de corte de proporción**, ilustraremos cómo se aplica para dividir un **gráfico completo**,  $K_5$ , con  $n = 5$  (sin bucles automáticos).

Ver [Fig. 6.11](#) donde las curvas rojas definen en cada caso qué nodos se incluyen en el subconjunto. Ay lo que pertenece a su complementario  $B = \bar{A}$ . Aunque teóricamente tenemos  $2^5 - 2$  particiones (descontando  $\emptyset$  y  $V$ ) para este gráfico, realmente tenemos 4 diferentes particiones: respectivamente con uno, dos, tres y cuatro elementos. De izquierda a derecha tenemos cuatro particiones.  $A_i: A_1 = \{3\}, A_2 = \{3, 1\}, A_3 = \{3, 1, 0\}$ , y  $A_4 = \{3, 1, 0, 2\}$ .

[Saltar al contenido principal](#)

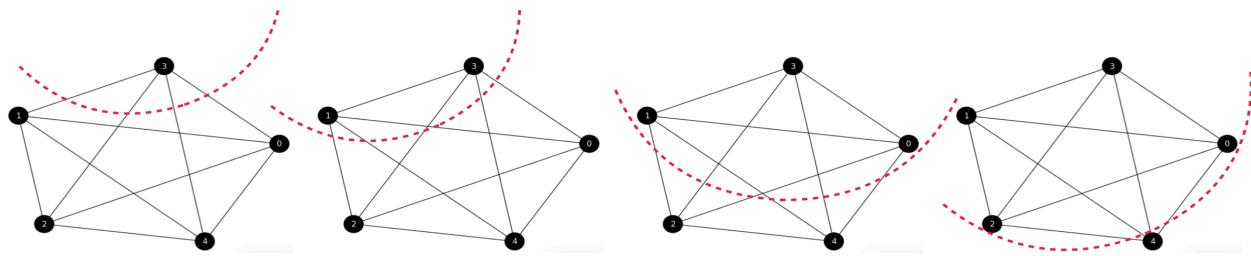


Fig. 6.11 De izquierda a derecha:  $(A_1, B_1), (A_2, B_2), (A_3, B_3)$  y  $(A_4, B_4)$ . Todos los cortes en  $K_n$  resultan de particiones que son igualmente óptimas!

Calculemos la pérdida de relación de corte para cada partición:

$$\begin{aligned} \text{cut}(A_1, B_1) &= 1 \cdot 4 \Rightarrow \text{Rcut}(A_1, B_1) = \frac{4}{|A_1|} + \frac{4}{|B_1|} = \frac{4}{1} + \frac{4}{4} = 5 \\ \text{cut}(A_2, B_2) &= 2 \cdot 3 \Rightarrow \text{Rcut}(A_2, B_2) = \frac{6}{|A_2|} + \frac{6}{|B_2|} = \frac{6}{2} + \frac{6}{3} = 5 \\ \text{cut}(A_3, B_3) &= 3 \cdot 2 \Rightarrow \text{Rcut}(A_3, B_3) = \frac{6}{|A_3|} + \frac{6}{|B_3|} = \frac{6}{3} + \frac{6}{2} = 5 \\ \text{cut}(A_4, B_4) &= 4 \cdot 1 \Rightarrow \text{Rcut}(A_4, B_4) = \frac{4}{|A_4|} + \frac{4}{|B_4|} = \frac{4}{1} + \frac{4}{4} = 5 \end{aligned}$$

Como resultado, ¡todos los recortes de proporciones son iguales! ¿Por qué?

Considere el caso general  $K_n = (V, E)$  con  $n$  nodos. Cada nodo en  $K_n$  está conectado a  $n - 1$  nodos (recuerde que hemos excluido los bucles automáticos).

Entonces si un subconjunto  $A_k \subset V$  tiene  $k$  nodos, tenemos eso  $B_k = V - A_k$  tienen  $n - k$  nodos. Dado que cada nodo en  $A_k$  está vinculado (en este gráfico) con cualquier nodo en  $B_k$  tenemos eso

$$\text{cut}(A_k, B_k) = k \cdot (n - k) .$$

lo que resulta en

$$\text{Rcut}(A_k, B_k) = \frac{k \cdot (n - k)}{|A_k|} + \frac{k \cdot (n - k)}{|B_k|} = \frac{k \cdot (n - k)}{k} + \frac{k \cdot (n - k)}{(n - k)} = n - k + k = n .$$

Luego, la relación de corte para todos los cortes en  $K_n$  es siempre.

### 6.2.3. El laplaciano y Courant-Fischer

**Uno de los descubrimientos** más interesantes de la teoría espectral es el de minimizar sólo necesitamos interpretarlo como el **valor propio** def que es un **vector propio** de una matriz simétrica llamada **laplaciana**.

El laplaciano de un grafo  $G = (V, E)$ , que denotamos como  $\Delta$ , es la matriz resultante de restar a la matriz de grados la matriz de adyacencia del grafo:

$$\Delta = \mathbf{D} - \mathbf{A} .$$

Es decir, en la diagonal de  $\Delta$  tenemos los grados  $d_u$  de todos los nodos  $u \in V$  y en fuera de la diagonal tenemos el negativo de las adyacencias  $-a_{uv}$ .

Para nuestro gráfico de ejemplo, en la Fig. 6.9 y la Fig. 6.10, tenemos:

[Saltar al contenido principal](#)

$$\Delta = \begin{bmatrix} \mathbf{d}_1 = 2 & -a_{12} = -1 & -a_{13} = -1 & -a_{14} = 0 & -a_{15} = 0 & -a_{16} = 0 \\ -a_{21} = -1 & \mathbf{d}_2 = 3 & -a_{23} = -1 & -a_{24} = -1 & -a_{25} = 0 & -a_{26} = 0 \\ -a_{31} = -1 & -a_{32} = -1 & \mathbf{d}_3 = 2 & -a_{34} = 0 & -a_{35} = 0 & -a_{36} = 0 \\ -a_{41} = 0 & -a_{42} = -1 & -a_{43} = 0 & \mathbf{d}_4 = 3 & -a_{45} = -1 & -a_{46} = -1 \\ -a_{51} = 0 & -a_{52} = 0 & -a_{53} = 0 & -a_{54} = -1 & \mathbf{d}_5 = 2 & -a_{56} = -1 \\ -a_{61} = 0 & -a_{62} = 0 & -a_{63} = 0 & -a_{64} = -1 & -a_{65} = -1 & \mathbf{d}_6 = 2 \end{bmatrix}$$

es decir

$$\Delta = \begin{bmatrix} \mathbf{2} & -1 & -1 & 0 & 0 & 0 \\ -1 & \mathbf{3} & -1 & -1 & 0 & 0 \\ -1 & -1 & \mathbf{2} & 0 & 0 & 0 \\ 0 & -1 & 0 & \mathbf{3} & -1 & -1 \\ 0 & 0 & 0 & -1 & \mathbf{2} & -1 \\ 0 & 0 & 0 & -1 & -1 & \mathbf{2} \end{bmatrix}$$

Algunas propiedades:

1. El laplaciano generalmente se considera el **operador derivativo** en el dominio del gráfico (es por eso que usamos la notación  $\Delta$ ) . En realidad para cualquier vector  $\mathbf{f} \in \mathbb{R}^{|V|}$  tenemos:

$$\Delta \mathbf{f} = \sum_{u \sim v} \mathbf{f}(u) - \mathbf{f}(v) .$$

En realidad, en el ejemplo anterior tenemos:

$$\Delta \mathbf{f} = \begin{bmatrix} \mathbf{2} & -1 & -1 & 0 & 0 & 0 \\ -1 & \mathbf{3} & -1 & -1 & 0 & 0 \\ -1 & -1 & \mathbf{2} & 0 & 0 & 0 \\ 0 & -1 & 0 & \mathbf{3} & -1 & -1 \\ 0 & 0 & 0 & -1 & \mathbf{2} & -1 \\ 0 & 0 & 0 & -1 & -1 & \mathbf{2} \end{bmatrix} \begin{bmatrix} \mathbf{f}(1) \\ \mathbf{f}(2) \\ \mathbf{f}(3) \\ \mathbf{f}(4) \\ \mathbf{f}(5) \\ \mathbf{f}(6) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(1) - \mathbf{f}(2) + \mathbf{f}(1) - \mathbf{f}(3) \\ \mathbf{f}(2) - \mathbf{f}(1) + \mathbf{f}(2) - \mathbf{f}(3) + \mathbf{f}(2) - \mathbf{f}(4) \\ \mathbf{f}(3) - \mathbf{f}(1) + \mathbf{f}(3) - \mathbf{f}(2) \\ \mathbf{f}(4) - \mathbf{f}(2) + \mathbf{f}(4) - \mathbf{f}(5) + \mathbf{f}(4) - \mathbf{f}(6) \\ \mathbf{f}(5) - \mathbf{f}(4) + \mathbf{f}(5) - \mathbf{f}(6) \\ \mathbf{f}(6) - \mathbf{f}(4) + \mathbf{f}(6) - \mathbf{f}(5) \end{bmatrix}$$

2. El laplaciano es un **operador armónico** , es decir, satisface que

$$\mathbf{f}(u) = \frac{1}{d_u} \sum_{v \sim u} f(v) .$$

y puedes comprobar esto en cualquier fila de  $\Delta \mathbf{f}$ .

3. El laplaciano está **vinculado con el pérdida** de la siguiente manera:

$$\mathbf{f}^T \Delta \mathbf{f} = \frac{1}{2} \sum_{u \sim v} (\mathbf{f}(u) - \mathbf{f}(v))^2$$

Para visualizar esto, consideremos el producto de  $\mathbf{f}^T$  y  $\Delta \mathbf{f}$  en el ejemplo:

$$[\mathbf{f}(1) \quad \mathbf{f}(2) \quad \mathbf{f}(3) \quad \mathbf{f}(4) \quad \mathbf{f}(5) \quad \mathbf{f}(6)] \begin{bmatrix} \mathbf{f}(1) - \mathbf{f}(2) + \mathbf{f}(1) - \mathbf{f}(3) \\ \mathbf{f}(2) - \mathbf{f}(1) + \mathbf{f}(2) - \mathbf{f}(3) + \mathbf{f}(2) - \mathbf{f}(4) \\ \mathbf{f}(3) - \mathbf{f}(1) + \mathbf{f}(3) - \mathbf{f}(2) \\ \mathbf{f}(4) - \mathbf{f}(2) + \mathbf{f}(4) - \mathbf{f}(5) + \mathbf{f}(4) - \mathbf{f}(6) \\ \mathbf{f}(5) - \mathbf{f}(4) + \mathbf{f}(5) - \mathbf{f}(6) \\ \mathbf{f}(6) - \mathbf{f}(4) + \mathbf{f}(6) - \mathbf{f}(5) \end{bmatrix}$$

de este producto de  $(1 \times |V|)(|V| \times 1)$  obtenemos un escalar de la siguiente manera:

[Saltar al contenido principal](#)

$$\mathbf{f}(1)\Delta\mathbf{f} + \mathbf{f}(2)\Delta\mathbf{f} + \dots + \mathbf{f}(|V|)\Delta\mathbf{f}.$$

y enchufando  $\Delta\mathbf{f} = \sum_{u \sim v} \mathbf{f}(u) - \mathbf{f}(v)$  tenemos

$$\mathbf{f}(1) \sum_{1 \sim v} \mathbf{f}(1) - \mathbf{f}(v) + \mathbf{f}(2) \sum_{2 \sim v} \mathbf{f}(2) - \mathbf{f}(v) + \dots + \mathbf{f}(n) \sum_{n \sim v} \mathbf{f}(n) - \mathbf{f}(v).$$

dónden  $= |V|$ . Esto lleva a

$$\mathbf{f}(1) \sum_{1 \sim v} \mathbf{f}(1)^2 - \mathbf{f}(1)\mathbf{f}(v) + \sum_{2 \sim v} \mathbf{f}^2(2) - \mathbf{f}(2)\mathbf{f}(v) + \dots + \sum_{n \sim v} \mathbf{f}(n)^2 - \mathbf{f}(n)\mathbf{f}(v).$$

Para cualquier ventaja, digamos  $e = (u, v)$  la suma anterior contiene  $f(u)^2 + f(v)^2 - 2f(u)f(v) = (f(u) - f(v))^2$ .

Ver por ejemplo,  $e = (1, 2)$  que involucra nodos 1 y 2. De la suma anterior (véanse también los dos primeros elementos de  $\Delta\mathbf{f}$ ) tenemos:

$$f(1)^2 - f(1)f(2) + f(1)^2 - f(1)f(3)$$

y

$$f(2)^2 - f(2)f(1) + f(2)^2 - f(2)f(3) + f(2)^2 - f(2)f(4)$$

Para  $e = (1, 2)$  nosotros tomamos

$$f(1)^2 - f(1)f(2) - f(2)f(1) + f(2)^2 = (f(1) - f(2))^2$$

y hay más términos para  $e = (2, 1)$  y de manera similar para los bordes restantes, lo que lleva a

$$2 \sum_{u \sim v} (f(u) - f(v))^2.$$

Por la identidad de Lagrange sabemos que

$$\sum_{u,v} (\mathbf{f}(u) - \mathbf{f}(v))^2 = 2|V| \sum_u \mathbf{f}(u)^2.$$

Como resultado tenemos que

$$\lambda = \left\{ \frac{\sum_{u \sim v} (\mathbf{f}(u) - \mathbf{f}(v))^2}{\sum_u \mathbf{f}(u)^2} : \mathbf{f} \neq \mathbf{0}, \mathbf{f} \neq c \cdot \mathbf{1}, c \in \mathbb{R} \right\}.$$

Desde  $\sum_u \mathbf{f}(u)^2 = \mathbf{f}^T \mathbf{f} = \|\mathbf{f}\|^2$  es el módulo cuadrado de  $\mathbf{f}$ , tenemos una definición más compacta de  $\lambda$ :

$$\lambda = \left\{ \frac{\mathbf{f}^T \Delta \mathbf{f}}{\mathbf{f}^T \mathbf{f}} : \mathbf{f} \neq \mathbf{0}, \mathbf{f} \neq c \cdot \mathbf{1}, c \in \mathbb{R} \right\}.$$

**Teorema de Courant-Fischer** . dado una  $n \times n$  matriz simétrica  $\Delta$ , Tienen valores propios reales  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  con sus vectores propios de valor real asociados  $\phi_1, \phi_2, \dots, \phi_n$  satisfactorio  $\Delta \phi_i = \lambda_i \phi_i$  también satisface:

[Saltar al contenido principal](#)

$$\begin{aligned}\lambda_1 &= \min_{\mathbf{f} \neq 0} \frac{\mathbf{f}^T \Delta \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \Rightarrow \mathbf{f} = \phi_1 \\ \lambda_2 &= \min_{\mathbf{f} \neq 0, \mathbf{f} \perp \phi_1} \frac{\mathbf{f}^T \Delta \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \Rightarrow \mathbf{f} = \phi_2 \\ \lambda_3 &= \min_{\mathbf{f} \neq 0, \mathbf{f} \perp \{\phi_1, \phi_2\}} \frac{\mathbf{f}^T \Delta \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \Rightarrow \mathbf{f} = \phi_3 \\ &\dots \\ \lambda_n &= \min_{\mathbf{f} \neq 0, \mathbf{f} \perp \{\phi_1, \phi_2, \dots, \phi_{n-1}\}} \frac{\mathbf{f}^T \Delta \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \Rightarrow \mathbf{f} = \phi_n\end{aligned}$$

Al observar los vectores propios, se interpretan como **funciones de valor real**.  $\phi : V \rightarrow \mathbb{R}$  y los valores propios son las variaciones de estas funciones a lo largo del gráfico  $G = (V, E)$ .

**Espectro laplaciano y vectores propios**. Del teorema anterior tenemos que para el laplaciano  $\Delta$ :

1.  $\lambda_1 = 0$ . Dado que las filas del laplaciano suman 0, el valor propio más pequeño del laplaciano es  $\lambda_1 = 0$  con vector propio  $\phi = \mathbf{1}$ .

$$\Delta \phi_1 = \lambda_1 \phi_1 \Rightarrow \Delta \mathbf{1} = 0 \cdot \mathbf{1} .$$

2. Si la gráfica es conexa,  $\lambda_2 = \min \lambda$  (mínimo de la **pérdida de Fiedler**). Como resultado,  $\lambda_2$ , el **valor de Fiedler** se puede interpretar como **la varianza mínima distinta de cero** que se puede lograr para funciones que son perpendiculares a  $\mathbf{1}$ , y el vector de Fiedler es esa función.

Por ejemplo, el vector de Fiedler para el ejemplo pequeño se representa en el gráfico de [la figura 6.12](#).

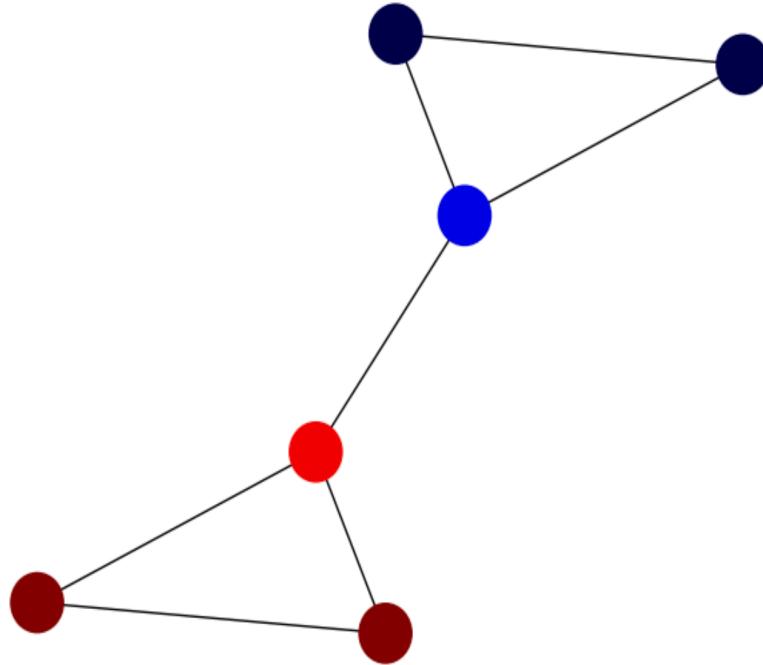


Fig. 6.12 Gráfico pequeño y su vector de Fiedler mapeado en los nodos.

En el gráfico anterior tenemos:

$$\lambda_2 = 0.438, \quad \phi_2^T = [-0.464 \ -0.260 \ -0.464 \ +0.260 \ +0.464 \ +0.464] ,$$

donde tenemos **parte de los componentes negativos** (correspondientes a  $-1$ ) y parte de ellos positivos (correspondientes a  $+1$ ), aproximando así una partición en el tiempo  $O(n^3)$ , el necesario para resolver un sistema.

Curiosamente los componentes  $\phi_2(2)$  y  $\phi_2(4)$  tienen el **valor absoluto más pequeño** aunque definen una ventaja heterófila. Ésta es una limitación de la relajación continua.

[Saltar al contenido principal](#)

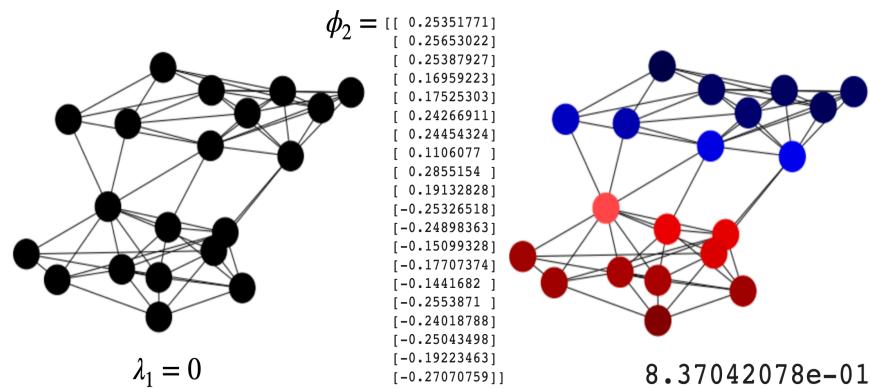


Fig. 6.13 SBM y su vector de Fiedler mapeado en los nodos.

**El espectro** . El espectro de  $\Delta$  es la colección de **valores propios**  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ . No son decrecientes porque resumen la variabilidad creciente de sus respectivos **vectores propios** mapeados en los nodos del gráfico (ver Fig. 6.14 ).

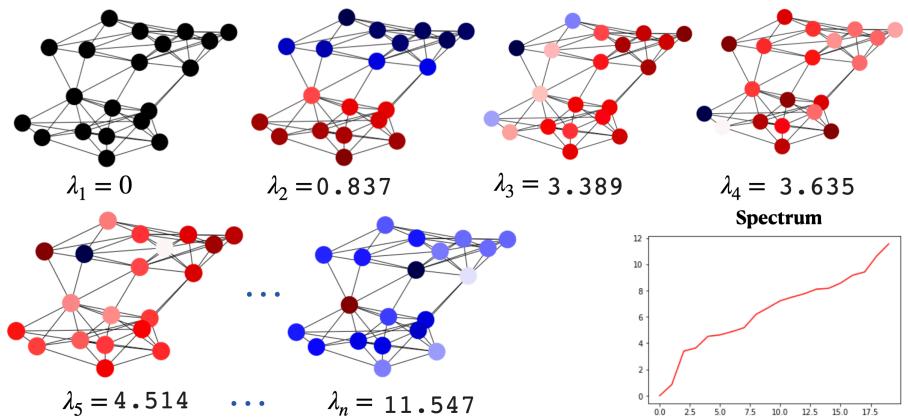


Fig. 6.14 Espectro SBM y sus vectores propios asociados.

## 6.2.4. El teorema

**El teorema espectral** . Uno de los hechos más interesantes de la teoría espectral en general es que, a pesar de que en la IA no se necesitan el espectro completo y los vectores propios, **cualquier matriz simétrica y cuadrada se puede descomponer** de la siguiente manera:

$$\Delta = \lambda_1 \phi_1 \phi_1^T + \lambda_2 \phi_2 \phi_2^T + \dots + \lambda_n \phi_n \phi_n^T .$$

Más compacto,

$$\Delta = \Phi \Phi^T = \sum_{i=1}^n \lambda_i \phi_i \phi_i^T ,$$

donde las columnas de  $\Phi$  son  $\phi_1, \phi_2, \dots, \phi_n$ .

El significado de este teorema es que la matriz ( $\Delta$  por ejemplo) se puede descomponer perfectamente (sin pérdida de información) como una **combinación lineal** de matrices  $\phi_i^T \phi_i$ , cada uno definido por un autovalor, y los coeficientes de esta combinación lineal son los coeficientes de los autovalores.

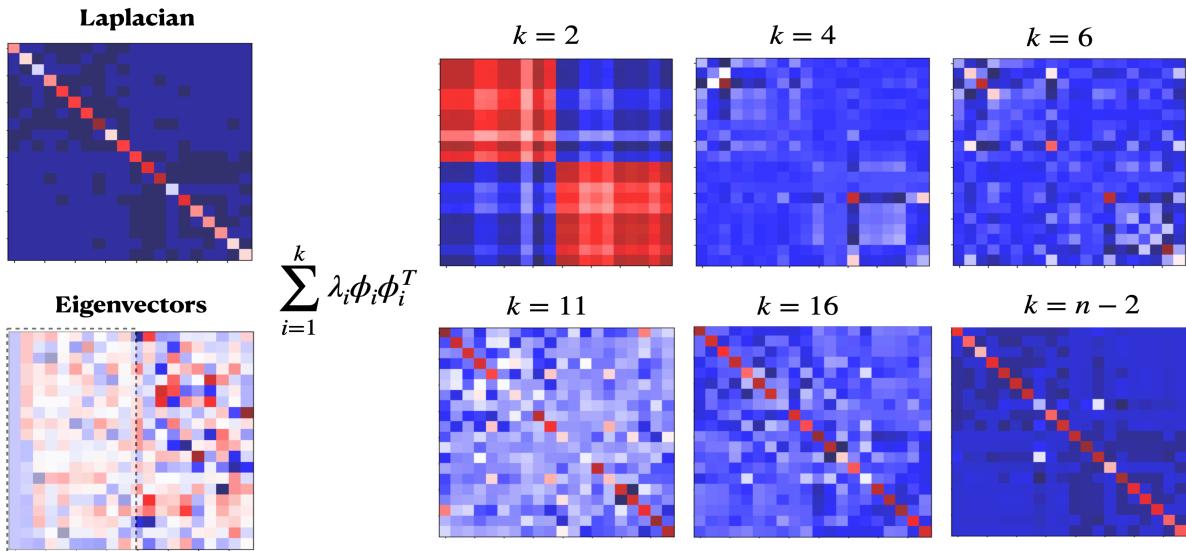
Sin embargo, si no tenemos el conjunto completo de vectores propios sino un pequeño número  $k < n$  de ellos, lo único que hacemos es **aproximar** la matriz (la laplaciana del gráfico en el caso de  $\Delta$ ). El **error de la aproximación** viene dado por la suma de los valores absolutos de los **valores propios o modas descartados** .

En la teoría espectral de grafos, es bastante común retener (o calcular) solo el valor más pequeño,  $k$  vectores propios del laplaciano  $\Delta$ :

[Saltar al contenido principal](#)

$$\Delta = \sum_{i=1}^k \lambda_i \phi_i \phi_i^T .$$

Considere, por ejemplo, el ejemplo de SBM desarrollado en este tema. En [la figura 6.15](#) mostramos la matriz laplaciana. $\Delta$  (arriba a la izquierda) y sus vectores propios  $\phi_1, \phi_2, \dots, \phi_n$  (columnas de izquierda a derecha). En [la figura 6.14](#) se muestra el mapeo de cada vector propio en el gráfico y el espectro laplaciano.



*Fig. 6.15 Vectores propios y reconstrucción de SBM.*

Luego, mirando la matriz de vectores propios  $\Phi$  (abajo a la izquierda), el primero  $\phi_1$  es constante **1**.

**Aproximación de primer orden**. Para  $k = 1$  tenemos la aproximación:

$$\Delta = \lambda_1 \phi_1 \phi_1^T = 0 \cdot \mathbf{1} \mathbf{1}^T = \mathbf{0} ,$$

es decir, la **aproximación de primer orden** del laplaciano es una matriz de unos (todos los elementos de la matriz son igualmente importantes) ponderada por cero. Sabemos que esto es muy incorrecto ya que a la diagonal se le deben asignar los grados del nodo. Esto es exactamente lo que el producto  $\lambda_1 = 0$ ; medio!

**Aproximación de segundo orden**. El segundo vector propio, la aproximación del vector de Fiedler, divide el gráfico en dos comunidades y es por eso que la aproximación de segundo orden  $k = 2$ :

$$\Delta = \lambda_1 \phi_1 \phi_1^T + \lambda_2 \phi_2 \phi_2^T$$

Capta **la información dominante** en el gráfico: las **dos comunidades en el SBM**. Recuerde que el rojo significa valor positivo alto y el azul negativo alto. Entonces para  $k = 2$  sólo sabemos que existe una alta probabilidad de que la mitad de los nodos pertenezcan a una comunidad y la otra mitad a otra.

**Aproximación de orden medio**. Para  $k > 2$  observamos que se capturan pocos elementos en la diagonal. Véase, por ejemplo, las esquinas rojas en  $k = 4$  y  $k = 6$ . Estos dos nodos pertenecen a bordes entre clases. También están vinculados con nodos de la misma comunidad: recuerde que la adyacencia en el laplaciano es  $-1$  (codificado en azul oscuro).

**Aproximación de alto orden**. Mira eso para  $k \geq 11$ ; La información de grado se recupera antes de que surja la adyacencia completa (negativa)!

Por lo tanto, el orden lógico de la información emergente durante la aproximación es:

1. Estructura de conglomerados del gráfico.
2. Nodos importantes (entre clases).
3. Información de grado.
4. Adyacencia negativa

[Saltar al contenido principal](#)

Para una mejor comprensión de estos pasos te proponemos un pequeño ejercicio.

**Ejercicio**. Consideremos el pequeño SBM cuyo vector de Fiedler se muestra en [la figura 6.12](#). Sólo tenemos  $\lambda_1, \phi_1$  y  $\lambda_2, \phi_2$ . Luego, de:

$$\begin{aligned}\lambda_1 &= 0.0 \quad \phi_1 = [+1.0 \ +1.0 \ +1.0 \ +1.0 \ +1.0 \ +1.0]^T \\ \lambda_2 &= 0.4 \quad \phi_2 = [-0.4 \ -0.3 \ -0.4 \ +0.3 \ +0.4 \ +0.4]\end{aligned}$$

Aplicar el teorema espectral para **aproximar la matriz laplaciana** utilizando la información proporcionada. Tenemos:

$$\Delta \approx \lambda_1 \phi_1 \phi_1^T + \lambda_2 \phi_2 \phi_2^T$$

Desde  $\lambda_1 = 0$ , tenemos:

$$\Delta \approx \lambda_2 \phi_2 \phi_2^T .$$

$$\begin{aligned}\phi_2 \phi_2^T &= \begin{bmatrix} -0.4 \\ -0.3 \\ +0.3 \\ +0.4 \\ +0.4 \end{bmatrix} \begin{bmatrix} -0.4 & -0.3 & -0.4 & +0.3 & +0.4 & +0.4 \end{bmatrix} \\ \phi_2 \phi_2^T &= \begin{bmatrix} 0.16 & 0.12 & 0.16 & -0.12 & -0.16 & -0.16 \\ 0.12 & 0.09 & 0.12 & -0.09 & -0.12 & -0.12 \\ 0.16 & 0.12 & 0.16 & -0.12 & -0.16 & -0.16 \\ -0.12 & -0.09 & -0.12 & 0.09 & 0.12 & 0.12 \\ -0.16 & -0.12 & -0.16 & 0.12 & 0.16 & 0.16 \\ -0.16 & -0.12 & -0.16 & 0.12 & 0.16 & 0.16 \end{bmatrix} \\ \lambda_2 \phi_2 \phi_2^T &= \begin{bmatrix} 0.064 & 0.048 & 0.064 & -0.048 & -0.064 & -0.064 \\ 0.048 & 0.036 & 0.048 & -0.036 & -0.048 & -0.048 \\ 0.064 & 0.048 & 0.064 & -0.048 & -0.064 & -0.064 \\ -0.048 & -0.036 & -0.048 & 0.036 & 0.048 & 0.048 \\ -0.064 & -0.048 & -0.064 & 0.048 & 0.064 & 0.064 \\ -0.064 & -0.048 & -0.064 & 0.048 & 0.064 & 0.064 \end{bmatrix}\end{aligned}$$

Luego, con los dos primeros valores propios y vectores propios todo lo que podemos hacer es capturar la estructura de bloques (comunidades) en el SBM, como se muestra en [la figura 6.15](#). ¡Mire que los bloques positivos (rojos en [la figura 6.15](#)) corresponden a los vínculos intraclase dentro de las dos comunidades, mientras que los bloques negativos corresponden a los bordes entre clases!

# Distancias y espacios latentes

## Contenido

- 7.1. Laplaciano normalizado
- 7.2. Tiempos de viaje e integración

### 7.1. Laplaciano normalizado

#### 7.1.1. Derivación

El laplaciano normalizado  $\tilde{\Delta}$  es una ligera modificación del laplaciano que fue diseñado inicialmente para relacionar los espectros de matrices de transición y matrices laplacianas. De hecho, los primeros capítulos del libro de [Fan Chung-Graham](#) son imprescindibles para los estudiantes interesados en la teoría de grafos espectrales, aunque el nivel es muy alto en términos matemáticos. Aquí nos referiremos a conceptos básicos para la lectura de los Alers.

**Derivación de la Transición**. Recuerde la matriz **de transición de  $G = (V, E)$** :

$$\mathbf{P} = \mathbf{D}^{-1} \mathbf{A} \text{ i.e. } p_{ij} = \frac{a_{ij}}{d_i} .$$

Si  $\mathbf{P}$  es irreducible y aperiódico (por ejemplo el dado por la matriz de Google), tenemos que el espectro (valores propios) de  $\mathbf{P}$  satisface:

$$1 = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > -1 .$$

Definamos ahora la siguiente matriz  $\tilde{\mathbf{P}}$  (que es diferente de  $\mathbf{P}$ ):

$$\tilde{\mathbf{P}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} ,$$

dónde  $\mathbf{D}^{-1/2}$  es la matriz diagonal con elementos  $1/\sqrt{d_i}$ .

Entonces, el **laplaciano normalizado** viene dado por

$$\tilde{\Delta} = \mathbf{I} - \tilde{\mathbf{P}} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} ,$$

dónde  $\mathbf{I}$  es la matriz identidad del tamaño  $|V|$ . Resulta que si  $\lambda_k$  es un valor propio de  $\mathbf{P}$ , entonces  $1 - \lambda_k$  es un valor propio de  $\tilde{\Delta}$  y viceversa .

**Derivación del laplaciano**. En su libro, Fan Chun aborda el laplaciano normalizado de una manera diferente. Consideremos ahora el laplaciano (no normalizado)

$$\Delta = \mathbf{D} - \mathbf{A} .$$

Definamos  $\tilde{\Delta}$  como la pre-multiplicación y post-multiplicación de  $\Delta$  por  $\mathbf{D}^{-1/2}$  (normalización de grados):

$$\begin{aligned} \tilde{\Delta} &= \mathbf{D}^{-1/2} \Delta \mathbf{D}^{-1/2} = \mathbf{D}^{-1/2} (\mathbf{D} - \mathbf{A}) \mathbf{D}^{-1/2} \\ &= \mathbf{D}^{-1/2} \mathbf{D} \mathbf{D}^{-1/2} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \\ &= \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \\ &= \mathbf{I} - \tilde{\mathbf{P}} . \end{aligned}$$

Entonces, la estructura **por componentes** de  $\tilde{\Delta}$  es

[Saltar al contenido principal](#)

$$\tilde{\Delta}_{ij} = \begin{cases} -\frac{1}{\sqrt{d_i}\sqrt{d_j}} & \text{if } (i, j) \in E \\ 1 & \text{if } i = j \text{ and } d_i \neq 0 \\ 0 & \text{otherwise,} \end{cases}$$

es decir, en la diagonal tenemos  $\tilde{\Delta}_{ii} = 1$  (normalizado) y en la fuera de la diagonal tendremos  $-\frac{1}{\sqrt{d_i d_j}}$ .

**Vínculo entre los vectores propios**. Si  $\phi$  y  $\tilde{\phi}$  son respectivamente vectores propios de  $\Delta$  y  $\tilde{\Delta}$ , están relacionados por las ecuaciones:  $\tilde{\phi} = \mathbf{D}^{1/2} \phi$  y  $\phi = \mathbf{D}^{-1/2} \tilde{\phi}$ :

$$\begin{aligned} \tilde{\Delta} \tilde{\phi} &= \lambda \tilde{\phi} \Rightarrow \mathbf{D}^{-1/2} \Delta \mathbf{D}^{-1/2} \tilde{\phi} = \lambda \tilde{\phi} \\ &\Rightarrow \mathbf{D}^{-1/2} \Delta \mathbf{D}^{-1/2} (\mathbf{D}^{1/2} \phi) = \lambda (\mathbf{D}^{1/2} \phi) \\ &\Rightarrow \mathbf{D}^{-1/2} \Delta \phi = \lambda \mathbf{D}^{1/2} \phi \\ &\Rightarrow \mathbf{D}^{1/2} \mathbf{D}^{-1/2} \Delta \phi = \mathbf{D}^{-1/2} \lambda \mathbf{D}^{1/2} \phi \\ &\Rightarrow \Delta \phi = \mathbf{D}^{1/2} \lambda \mathbf{D}^{1/2} \phi \\ &\Rightarrow \Delta \phi = \lambda \mathbf{D} \phi. \end{aligned}$$

En otras palabras, los vectores propios de  $\tilde{\Delta}$  son los **vectores propios generalizados** de  $\Delta$ .

Por ejemplo, como  $\mathbf{1}$  es el vector propio de  $\Delta$  con valor propio  $\lambda_1 = 0$ , entonces  $\mathbf{D}^{1/2} \mathbf{1}$  es el vector propio correspondiente para  $\tilde{\Delta}$ . Esto significa que [el vector de Fiedler asociado con el laplaciano normalizado debe ser perpendicular a  \$\mathbf{D} \mathbf{1}\$](#) .

## 7.1.2. Propiedades espectrales de los gráficos

En general, usando el laplaciano normalizado obtenemos estimadores más robustos y es por eso que algunas propiedades del gráfico se pueden identificar simplemente mirando el espectro de  $\tilde{\Delta}$  son enunciados.

Dejarn = |V| en  $G = (V, E)$ . Entonces:

1. **Suma de espectros**. La suma de todos los valores propios es  $\sum_i \lambda_i \leq n$ , con igualdad sólo si la gráfica es conexa. La prueba es trivial ya que la traza de la matriz (suma de la diagonal) satisface  $\text{Tr}[\tilde{\Delta}] = \sum_i \lambda_i$ .
2. **Límite de los espectros**. Para todos  $i \leq n$  tenemos  $\lambda_i \leq 2$ , con  $\lambda_n = 2$  sólo si un componente conexo del gráfico es bipartito.
3. **Extensión de los espectros**. Un gráfico sin nodos aislados satisface  $\lambda_n \geq \frac{n}{n-1}$  con igualdad si el gráfico está completo (sin auto-bucle).
4. **Número de componentes conectados**. Hay  $k$  componentes conectados si  $\lambda_1 = \lambda_2 = \dots = \lambda_k = 0$ .

**Conectividad algebraica**  $\lambda_2$ . El segundo valor propio más pequeño de  $\tilde{\Delta}$  (el valor de Fiedler) es uno de los más informativos ya que:

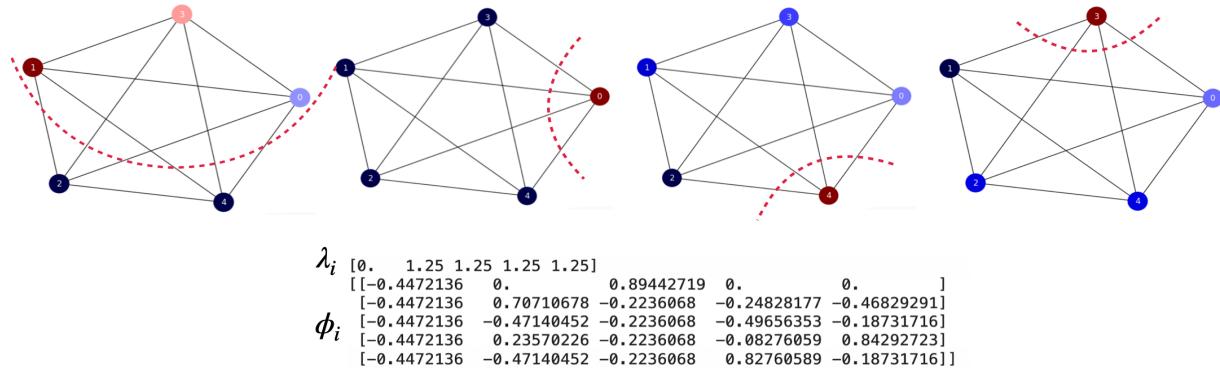
5.  $\lambda_2 \leq \frac{n}{n-1}$  con igualdad sólo si  $G$  es el gráfico completo (sin auto-bucle).
6.  $\lambda_2 < 1$  si el  $G$  no es completo.
7. El diámetro  $D$  de  $G$  (longitud del camino más corto entre cualquier par de nodos) satisface:  $\frac{1}{D \text{vol} G} < \lambda_2$ .

Algunos **ejemplos de espectros gráficos** (del libro de Chung):

- **gráfico completo**  $K_n$  (sin autobucle): es 0 y  $\frac{n}{n-1}$  (con multiplicidad  $n-1$ ).
- **Gráfico estelar (centro)**  $S_n$ : es 0, 1 (con multiplicidad  $n-2$ ) y 2.
- **Gráfico de ruta**  $P_n$ : es  $1 - \cos \frac{\pi k}{n-1}$  para  $k = 0, 1, \dots, n-1$ .
- **Gráfico de ciclo**  $C_n$ : es  $1 - \cos \frac{2\pi k}{n}$  para  $k = 0, 1, \dots, n-1$ .

**El espectro del gráfico completo**. [Figura 7.1](#). Hemos analizado la solución óptima de ratio-cut para este gráfico en el tema anterior y nos dimos cuenta de que **todos los cortes tienen el mismo coste**. Esta es la razón por la que el espectro no trivial es  $\lambda_i = \frac{n}{n-1}$ ,  $i \geq 1$  (para  $n = 5$  es  $\lambda_i = 1.25$ ,  $i \geq 1$ ). Los vectores propios no triviales corresponden a diferentes soluciones igualmente óptimas.

[Saltar al contenido principal](#)

Figura 7.1  $K_5$ . Espectro de Laplaciano normalizado y vectores propios no triviales.

**El espectro del gráfico estelar** . Figura 7.2 . Para el gráfico de estrellas, podemos analizar un poco la solución óptima en términos de la relación de pérdida de corte. Para hacer eso, por  $S_5$  (que tienen = 5 nodos) generemos particiones con 0, 1, 2, 3 Nudos del mismo color que el central.

- Cuando **solo el nodo central** tiene un color diferente a los restantes tenemos  $A_1 = \{0\}, B_1 = \{1, 2, 3, 4\}$  y

$$\text{Rcut}(A_1, B_1) = \frac{\text{cut}(A_1, B_1)}{|A_1|} + \frac{\text{cut}(A_1, B_1)}{|B_1|} = \frac{4}{1} + \frac{4}{4} = 4 + \frac{4}{4} = 5 .$$

- Cuando **el nodo central y 1 más** tienen el mismo color que los restantes tenemos, por ejemplo,  $A_2 = \{0, 1\}, B_2 = \{2, 3, 4\}$ . Entonces,  $\text{cut}(A_2, B_2) = |B_2| = 3$ . Entonces

$$\text{Rcut}(A_2, B_2) = \frac{|B_2|}{|A_1|} + \frac{|B_2|}{|B_1|} = \frac{3}{2} + \frac{3}{3} = 1 + \frac{3}{2} = \frac{5}{2} = 2.5 .$$

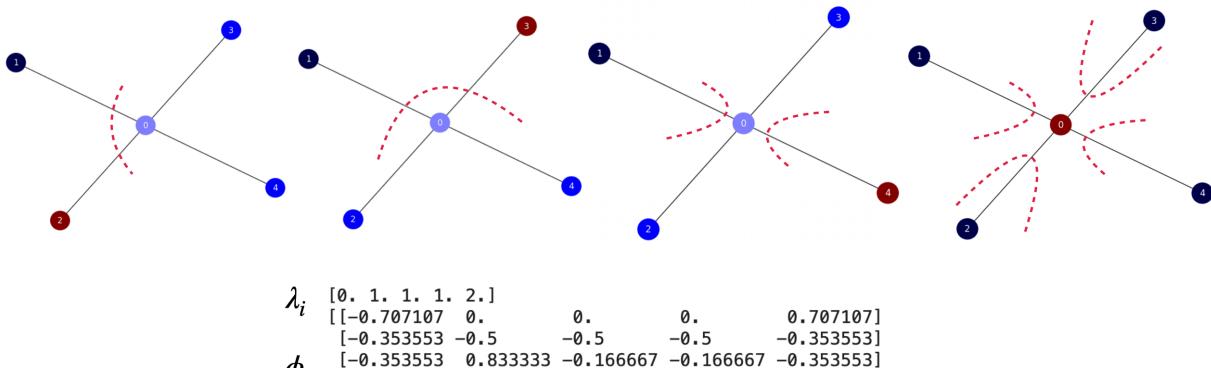
- Cuando **el nodo central y otros 2 nodos** tienen el mismo color que los restantes tenemos, por ejemplo,  $A_3 = \{0, 1, 2\}, B_3 = \{3, 4\}$ . Entonces,  $\text{cut}(A_3, B_3) = |B_3| = 2$ . Entonces

$$\text{Rcut}(A_3, B_3) = \frac{|B_3|}{|A_3|} + \frac{|B_3|}{|B_3|} = \frac{2}{3} + \frac{2}{2} = 1 + \frac{2}{3} = \frac{5}{2} = 2.5 .$$

- Cuando **el nodo central y otros 3 nodos** tienen el mismo color que los restantes tenemos, por ejemplo,  $A_4 = \{0, 1, 2, 3\}, B_4 = \{4\}$ . Entonces,  $\text{cut}(A_4, B_4) = |B_4| = 1$ . Entonces

$$\text{Rcut}(A_4, B_4) = \frac{|B_4|}{|A_4|} + \frac{|B_4|}{|B_4|} = \frac{1}{4} + \frac{1}{1} = 1 + \frac{1}{4} = \frac{5}{4} = 1.25 .$$

Por lo tanto, la mejor partición de corte de proporción es  $(A_4, B_4)$  (4 nodos del mismo color y 1 diferente). Aquí es fundamental que uno de los nodos del mismo color sea el central. De lo contrario tenemos la peor partición  $(A_1, B_1)$ . Las otras dos particiones tienen el mismo coste y son soluciones intermedias entre las mejores y las peores particiones.



[Saltar al contenido principal](#)

Figura 7.2  $S_5$ . Espectro de Laplaciano normalizado y vectores propios no triviales.

Sin embargo, **el enfoque espectral pierde la solución óptima**. ( $A_4, B_4$ ). El vector de Fiedler (primero a la izquierda en [la figura 7.2](#)) es básicamente una permutación del segundo y tercer vector propio. Todos estos vectores propios tienen el mismo valor propio: 1; codifican la misma solución: la que es igual a ( $A_2, B_2$ ) lo cual es **subóptimo** ya que su costo es 2.5.

Finalmente, el último vector propio, cuyo valor propio 2 codifica la peor solución ( $A_1, B_1$ )

A pesar de no encontrar la solución óptima, dado que la teoría espectral proporciona **aproximaciones**, este es un buen ejemplo de cómo la teoría espectral (que es una herramienta matemática continua) captura la naturaleza combinatoria de la coloración de gráficos (que es un problema discreto).

**El espectro del gráfico de ruta**. Este gráfico también ilustró la naturaleza combinatoria de la teoría espectral. En [la Fig. 7.3](#), mostramos el espectro y los vectores propios no triviales de  $P_5$ .

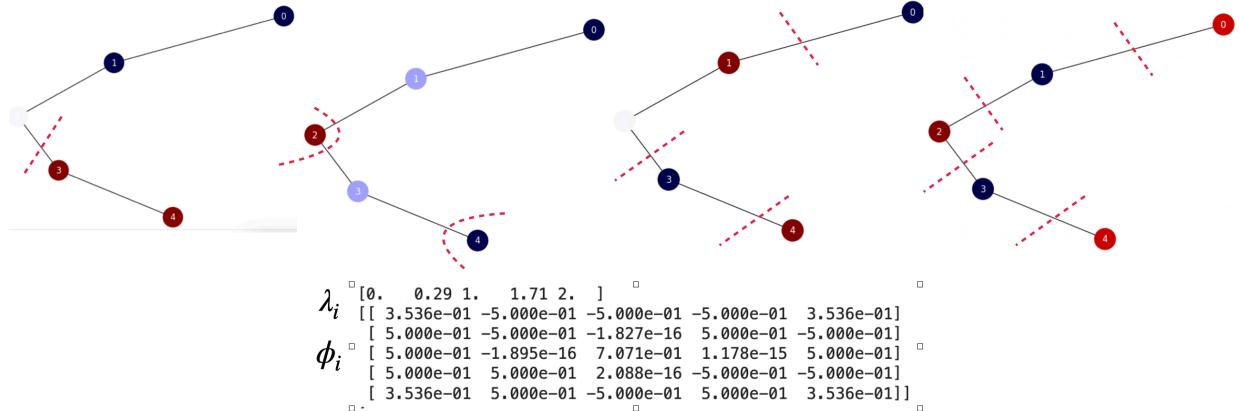
En términos combinatorios, vea que en un gráfico de ruta de  $n$  nodos que tenemos  $n - 1$  bordes. Como de costumbre, tenemos  $2^n - 2$  maneras (eliminando  $\emptyset$  y  $V$ ) de biparticionar este gráfico.

Dado que el ratio-cut prefiere cortes equilibrados, las soluciones óptimas (puede haber más de una debido a la simetría de las gráficas) corresponden a tener  $\text{cut}(A_i, B_i) = 1$  y  $|A_i| = \lceil n/2 \rceil$  (techo de la división de números enteros). Como resultado

$$\text{Rcut}(A_i, B_i) = \frac{1}{\lceil n/2 \rceil} + \frac{1}{n - \lceil n/2 \rceil} .$$

Para  $P_5$ , desdén = 5 es raro entonces  $\lceil n/2 \rceil = \lceil 5/2 \rceil = 3$  y la relación óptima de pérdida de corte es:

$$\text{Rcut}(A_1, B_1) = \frac{1}{3} + \frac{1}{5 - 3} = \frac{1}{3} + \frac{1}{2} = \frac{5}{6} = 0.83 .$$

Figura 7.3  $P_5$ . Espectro de Laplaciano normalizado y vectores propios no triviales.

En este caso, la aproximación espectral devuelve la solución óptima. ( $A_1, B_1$ ) (ver el vector de Fiedler en la figura, con valor propio 0.29). Los valores propios restantes son mayores o iguales que 1 y codificar soluciones cada vez más no óptimas.

Por ejemplo, la solución codificada por el tercer vector propio más pequeño del laplaciano normalizado es  $A_2 = \{0, 1, 3\}$  y  $B_2 = \{2, 4\}$  y tiene  $\text{cut}(A_2, B_2) = 3$ . Entonces:

$$\text{Rcut}(A_2, B_2) = \frac{3}{3} + \frac{3}{2} = 1 + \frac{3}{2} = \frac{5}{2} = 2.5 .$$

El correspondiente al cuarto vector propio más pequeño es  $A_3 = \{0, 3\}$ ,  $B_3 = \{1, 2, 4\}$  con  $\text{cut}(A_3, B_3) = 3$  y

$$\text{Rcut}(A_3, B_3) = \frac{3}{2} + \frac{3}{3} = 1 + \frac{3}{2} = \frac{5}{2} = 2.5 .$$

[Saltar al contenido principal](#)

Finalmente, el vector propio con el valor propio más grande corresponde a  $A_4 = \{0, 2, 4\}$  y  $B_4 = \{1, 3\}$  con el corte más grande:  $\text{cut}(A_3, B_3) = 4y$

$$\text{Rcut}(A_4, B_4) = \frac{4}{3} + \frac{4}{2} = \frac{4}{3} + \frac{4}{2} = \frac{8+12}{6} = \frac{20}{6} = 3.3 .$$

Por lo tanto, en este caso la teoría espectral califica las soluciones discretas exactamente.

**El espectro del gráfico del ciclo**. Finalmente, el gráfico Ciclo es una variante del gráfico Ruta donde el nodo final y el inicial están vinculados. Esto simplifica mucho el análisis. Como podemos ver en [la Fig. 7.4](#), el vector de Fiedler codifica la solución óptima.  $A_1 = \{0, 1, 4\}$ ,  $B_1 = \{2, 3\}$ ,  $\text{concut}(A_1, B_1) = 2$ . Entonces

$$\text{Rcut}(A_1, B_1) = \frac{2}{3} + \frac{2}{2} = 1 + \frac{2}{3} = \frac{5}{3} = 1.67 .$$

En realidad, dado que el segundo y el tercer valor propio son iguales, sus vectores propios se aproximan a la misma solución. La solución dada por el tercer vector propio es  $A_2 = \{0, 3, 4\}$  y  $B_2 = \{1, 2\}$   $\text{concut}(A_2, B_2) = 2$  y

$$\text{Rcut}(A_2, B_2) = \frac{2}{3} + \frac{2}{2} = 1 + \frac{2}{3} = \frac{5}{3} = 1.67 .$$

El siguiente vector propio codifica  $A_3 = \{0, 2, 3\}$ ,  $B_3 = \{1, 4\}$   $\text{concut}(A_3, B_3) = 4y$

$$\text{Rcut}(A_3, B_3) = \frac{4}{3} + \frac{4}{2} = \frac{8+12}{6} = \frac{20}{6} = 3.3 .$$

Finalmente, el último vector propio más pequeño codifica la solución.  $A_4 = \{0, 2, 4\}$  y  $B_4 = \{1, 3\}$   $\text{concut}(A_4, B_4) = 4y$

$$\text{Rcut}(A_4, B_4) = \frac{4}{3} + \frac{4}{2} = \frac{8+12}{6} = \frac{20}{6} = 3.3 .$$

En realidad, los dos últimos vectores propios tienen los mismos valores propios y, por lo tanto, se aproximan a la misma solución.

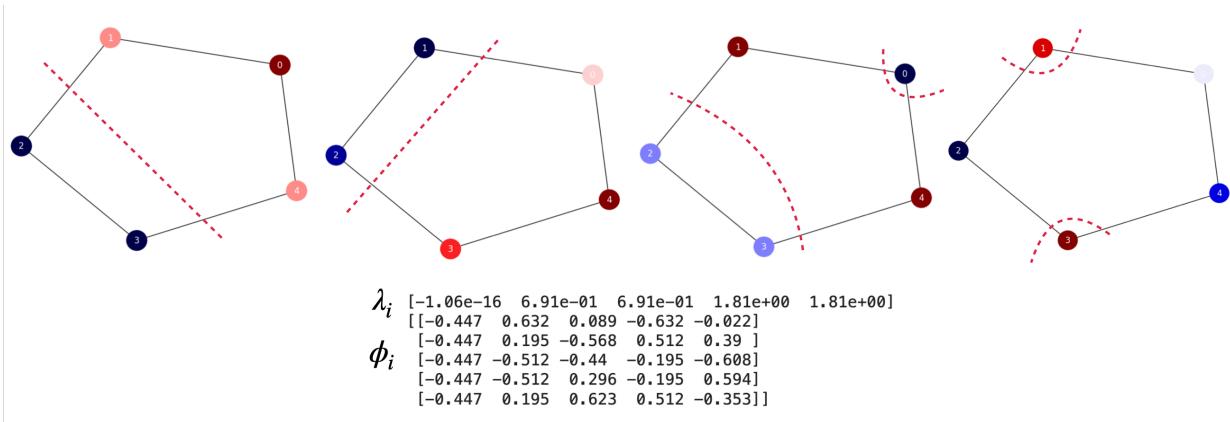


Figura 7.4C5. Espectro de Laplaciano normalizado y vectores propios no triviales.

### 7.1.2.1. La constante Cheeger

El valor de Fiedler  $\lambda_2$  se utiliza frecuentemente para **cuantificar qué tan débil es el gráfico** (cuanto más pequeño  $\lambda_2$  el más débil  $G$ ). Esto es particularmente interesante cuando se limita la llamada **constante de Cheeger**. Lo definimos de la siguiente manera:

Considere una partición  $V = A \cup \bar{A}$ ,  $A \cap \bar{A} = \emptyset$ , entonces la *conductancia* de la partición es

[Saltar al contenido principal](#)

$$h(A) = \frac{\text{cut}(A, \bar{A})}{\min\{|A|, |\bar{A}|\}} ,$$

y la constante de Cheeger es la \*conductancia mínima para cualquier partición de  $V$  (definido como por subconjunto  $A \subseteq V$  y su complementario  $\bar{A} = V - A$ ):

$$h_G = \min_{A \subseteq V} h(A) .$$

Dado que parece que minimizar la constante de Cheeger es bastante similar a minimizar  $\text{Rcut}(A, \bar{A})$ , no es sorprendente que calcular la constante de Cheeger sea NP-difícil (debemos evaluar todas las posibles  $2^{|V|}$  particiones/subconjuntos de  $V$ ). Sin embargo, dado que tenemos el valor de Fiedler, que es  $\lambda_2 > 0$  para un gráfico conexo, tenemos el siguiente límite fundamental de la teoría de grafos espectrales:

$$2h_G \geq \lambda_2 \geq \frac{h_G^2}{2} .$$

La prueba se puede encontrar en el libro de Chung (Capítulo ~2). De todos modos, este límite es interesante ya que nos dice qué tan precisa es la aproximación de la constante de Cheeger usando  $\lambda_2$  (también conocida como **brecha espectral**). Cuanto más amplio sea el cuello de botella del gráfico (conductancia grande), peor será la aproximación.

### 7.1.2.2. El tiempo de mezcla

La **brecha espectral**  $\lambda_2$  también es clave para caracterizar el tiempo de mezcla de un gráfico. Recuerde que el tiempo de mezcla de una matriz de transición  $\mathbf{P}$  asociado con un gráfico  $G = (V, E)$  es el tiempo necesario para alcanzar la distribución en estado estacionario  $\pi$  donde la probabilidad de ir desde un nodo  $j$  a cualquier otro es proporcional al grado del destino:  $\pi_i = \frac{d_j}{\text{vol}(G)}$ .

Para explicar el vínculo entre la brecha espectral y el tiempo de mezcla, seguimos algunos pasos formales:

1. **Vincular las matrices  $\mathbf{P}$  y  $\tilde{\mathbf{P}}$** .
2. **Vincular los vectores propios de  $\mathbf{P}$  y  $\tilde{\mathbf{P}}$** .
3. **Aplicar el teorema espectral**.

**Vincula las matrices**. las matrices  $\mathbf{P} = \mathbf{D}^{-1} \mathbf{A} \mathbf{y} \tilde{\mathbf{P}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$  están relacionados de la siguiente manera:

$$\begin{aligned} \tilde{\mathbf{P}} &= \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \Rightarrow \mathbf{D}^{-1/2} \tilde{\mathbf{P}} = \mathbf{D}^{-1/2} \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \\ &\Rightarrow \mathbf{D}^{-1/2} \tilde{\mathbf{P}} = \mathbf{D}^{-1} \mathbf{A} \mathbf{D}^{-1/2} \\ &\Rightarrow \mathbf{D}^{-1/2} \tilde{\mathbf{P}} \mathbf{D}^{1/2} = \mathbf{D}^{-1} \mathbf{A} \mathbf{D}^{-1/2} \mathbf{D}^{1/2} \\ &\Rightarrow \mathbf{D}^{-1/2} \tilde{\mathbf{P}} \mathbf{D}^{1/2} = \mathbf{D}^{-1} \mathbf{A} \\ &\Rightarrow \mathbf{D}^{-1/2} \tilde{\mathbf{P}} \mathbf{D}^{1/2} = \mathbf{P} \end{aligned}$$

es decir  $\mathbf{P} = \mathbf{D}^{-1/2} \tilde{\mathbf{P}} \mathbf{D}^{1/2}$ .

**Vincular los vectores propios**. Si  $\phi$  es un vector propio de  $\mathbf{P}$  con valor propio  $\lambda$ , entonces  $\tilde{\phi} = \mathbf{D}^{1/2} \phi$  es un vector propio de  $\tilde{\mathbf{P}}$  con el mismo valor propio  $\lambda$ :

$$\begin{aligned} \mathbf{P} \phi &= \lambda \phi \Rightarrow \mathbf{D}^{-1/2} \tilde{\mathbf{P}} \mathbf{D}^{1/2} \phi = \lambda \phi \\ &\Rightarrow \mathbf{D}^{1/2} \mathbf{D}^{-1/2} \tilde{\mathbf{P}} \mathbf{D}^{1/2} \phi = \lambda \mathbf{D}^{1/2} \phi \\ &\Rightarrow \tilde{\mathbf{P}} \mathbf{D}^{1/2} \phi = \lambda \mathbf{D}^{1/2} \phi \\ &\Rightarrow \tilde{\mathbf{P}} \tilde{\phi} = \lambda \tilde{\phi} \end{aligned}$$

**Aplicar el teorema espectral**. En primer lugar, aplicamos el teorema espectral a  $\tilde{\mathbf{P}}$ :

$$\tilde{\mathbf{P}} = \sum_{i=1}^n \lambda_i \tilde{\phi}_i \tilde{\phi}_i^T ,$$

[Saltar al contenido principal](#)

Luego, siguiendo [Random Walks on Graphs: A Survey](#) de Lovasz, conectamos la expansión anterior  $\mathbf{P} = \mathbf{D}^{-1/2} \tilde{\mathbf{P}} \mathbf{D}^{1/2}$ :

$$\mathbf{P} = \mathbf{D}^{-1/2} \tilde{\mathbf{P}} \mathbf{D}^{1/2} = \sum_{i=1}^n \lambda_i \mathbf{D}^{-1/2} \tilde{\phi}_i \tilde{\phi}_i^T \mathbf{D}^{1/2}.$$

Ahora, ampliamos la expresión anterior de la siguiente manera:

$$\mathbf{P} = \lambda_1 \mathbf{D}^{-1/2} \tilde{\phi}_1 \tilde{\phi}_1^T \mathbf{D}^{1/2} + \sum_{i=2}^n \lambda_i \mathbf{D}^{-1/2} \tilde{\phi}_i \tilde{\phi}_i^T \mathbf{D}^{1/2}.$$

Desde  $\lambda_1 = 1$  y  $\tilde{\phi}_1 = \mathbf{D}^{1/2} \mathbf{1}$ , el primer término de la expresión anterior se convierte en:

$$\begin{aligned} 1 \cdot \mathbf{D}^{-1/2} \tilde{\phi}_1 \tilde{\phi}_1^T \mathbf{D}^{1/2} &= \mathbf{D}^{-1/2} \mathbf{D}^{1/2} \mathbf{1} (\mathbf{D}^{1/2} \mathbf{1})^T \mathbf{D}^{1/2} \\ &= \mathbf{1} \mathbf{1}^T \mathbf{D}^{1/2} \mathbf{D}^{1/2} \\ &= \mathbf{1} \mathbf{1}^T \mathbf{D} \\ &= \mathbf{1} \mathbf{1}^T \mathbf{D} \end{aligned}$$

donde definimos  $\mathbf{Q} = \mathbf{1} \mathbf{1}^T \mathbf{D}$  es una matriz cuyas filas se repiten y son iguales a

$$\mathbf{Q}_{i:} = [d_1 \ d_2 \ \dots \ d_n],$$

Recuerda (del tema anterior) que

$$\lim_{t \rightarrow \infty} \mathbf{P}^t = \Pi$$

dónde

$$\Pi_{i:} = \pi = \left[ \frac{d_1}{\text{vol}(G)} \ \frac{d_2}{\text{vol}(G)} \ \dots \ \frac{d_n}{\text{vol}(G)} \right],$$

es decir, toda la distribución estacionaria se repite en cada fila de  $\Pi$ .

Como resultado,  $\Pi$  es solo una **versión normalizada** de  $\mathbf{Q}$ , es decir, podemos interpretar  $\mathbf{Q}_{i:}$  como una **distribución de grados no normalizada**. Esto es exactamente todo lo que necesitamos para la prueba ya que tenemos:

$$\begin{aligned} \mathbf{P} &= \mathbf{Q} + \sum_{i=2}^n \lambda_i \mathbf{D}^{-1/2} \tilde{\phi}_i \tilde{\phi}_i^T \mathbf{D}^{1/2} \\ &\approx \Pi + \sum_{i=2}^n \lambda_i \mathbf{D}^{-1/2} \tilde{\phi}_i \tilde{\phi}_i^T \mathbf{D}^{1/2} \\ &\approx \Pi + \sum_{i=2}^n \lambda_i \mathbf{D}^{-1/2} \mathbf{R}_i \mathbf{D}^{1/2}, \end{aligned}$$

Ahora, cada término  $\tilde{\phi}_i \tilde{\phi}_i^T$  es una  $n \times n$  matriz  $\mathbf{R}_i$  donde  $\mathbf{R}_i(u, v) = \tilde{\phi}_i(u) \tilde{\phi}_i(v)$ .

Como resultado, tomar los componentes  $p_{uv}$  de  $\mathbf{P}$ , tenemos:

$$p_{uv} \approx \pi_{uv} + \sum_{i=2}^n \lambda_i \tilde{\phi}_i(u) \tilde{\phi}_i(v) \sqrt{\frac{d_v}{d_u}}.$$

Finalmente, aplicando el **problema espectral** a la potencia  $\mathbf{P}^t$  tenemos

$$\mathbf{P}^t = \sum_{i=1}^n \lambda_i^t \phi_i \phi_i^T,$$

[Saltar al contenido principal](#)

es decir, cualquier **función matricial** (por ejemplo, la potencia, exponencial, logaritmo) se **transfiere al espectro**.

Usando este resultado, tenemos que

$$p_{uv}^t \approx \pi_{uv} + \sum_{i=2}^n \lambda_i^t \tilde{\phi}_i(u) \tilde{\phi}_i(v) \sqrt{\frac{d_v}{d_u}}.$$

Lo que lleva a

$$p^t(v) \approx \pi(v) + \sum_{i=2}^n \lambda_i \tilde{\phi}_i(u) \tilde{\phi}_i(v) \sqrt{\frac{d_v}{d_u}}.$$

Entonces, la **tasa de convergencia** de Páles dado por

$$\begin{aligned} p^t(v) - \pi(v) &\approx \sum_{i=2}^n \lambda_i \tilde{\phi}_i(u) \tilde{\phi}_i(v) \sqrt{\frac{d_v}{d_u}} \\ |p^t(v) - \pi(v)| &\approx \sum_{i=2}^n \lambda_i \tilde{\phi}_i(u) \tilde{\phi}_i(v) \sqrt{\frac{d_v}{d_u}} \\ &< |\lambda_2^t| \sum_{i=2}^n \tilde{\phi}_i(u) \tilde{\phi}_i(v) \sqrt{\frac{d_v}{d_u}} \\ &< |\lambda_2^t| \cdot \sqrt{\frac{d_v}{d_u}}. \end{aligned}$$

donde la última simplificación se debe a que  $\sum_{i=2}^n \tilde{\phi}_i(u) \tilde{\phi}_i(v) \leq 1$  ya que los vectores propios son ortonormales:  $\tilde{\Phi} \tilde{\Phi}^T = \mathbf{I}$ .

Entonces, la tasa de convergencia a la distribución estable es rápida si la brecha espectral es pequeña.

## 7.2. Tiempos de viaje e

### 7.2.1. ¿Dónde viven los datos?

Considere un conjunto de puntos  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ , con  $x_i \in 3D$ . La métrica natural en  $3D$  es la distancia euclídea  $\|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^3 (\mathbf{x}_i(k) - \mathbf{x}_j(k))^2}$ . Véase, por ejemplo, la nube de puntos de la figura 7.5.

Esta nube de puntos (llamada **rollo suizo**) es la representación típica de datos como imágenes en espacios de baja dimensión (un punto por imagen) llamados **variedades**. Es bien sabido que estos espacios tienen una geometría curva. Imágenes similares, por ejemplo, son bastante parecidas en un espacio de baja dimensión, pero la semántica de tal similitud apenas es consistente con los grupos casi esféricos de los espacios euclidianos.

Como resultado, cuando intentamos agrupar estos datos, que viven en un espacio curvo, usando distancias euclidianas (ver Fig. 7.5 -izquierda donde configuramos  $C = 3$  clusters), los puntos lejanos en la variedad curva están agrupados.

Alternativamente, seguimos el enfoque propuesto en el artículo fundamental de [Tenenbaum et al.](#):

1. Creamos un gráfico  $G = (V, E)$  con  $|V| = n$ , donde hay una ventaja  $e = (i, j)$  si  $j$  es uno de  $K = 10$  vecinos más cercanos de  $i$ . Esto se llama **gráfico KNN o gráfico de vecinos más cercanos**. Bueno, las distancias euclidianas  $\|\mathbf{x}_i - \mathbf{x}_j\|$  solo se utilizan para construir el gráfico KNN.
2. Después de crear el gráfico KNN, calculamos los tres vectores propios  $\phi_2, \phi_3$  y  $\phi_4$  correspondiente a los tres valores propios más pequeños no triviales  $\lambda_2, \lambda_3$  y  $\lambda_4$  del laplaciano.
3. Es bien sabido que la **codificación** (también llamada **incrustación**) de cada  $\mathbf{x}_i$  como

$$\mathbf{z}_i = [\phi_2(i) \ \phi_3(i) \ \phi_4(i)]^T$$

Minimiza la pérdida de corte de relación del gráfico KNN  $G = (V, E)$ . Como resultado, cuando se utiliza la distancia euclídea entre  $\mathbf{z}_i - \mathbf{z}_j$  en el **espacio transformado o latente**, lograremos un agrupamiento consistente con el espacio curvo (ver

[Saltar al contenido principal](#)

En otras palabras, como hemos explicado en el tema anterior, la partición espectral conduce a agrupar los datos de una manera que sea consistente con la **geometría curva del gráfico**.

Pero, ¿cómo sabemos esto? Porque Tenenbaum et al. encontró que las distancias  $\|\mathbf{z}_i - \mathbf{z}_j\|$  están **correlacionados estadísticamente** con las distancias de camino más cortas SP entre los nodos, encontradas por Dijkstra o  $A^*$  algoritmos!

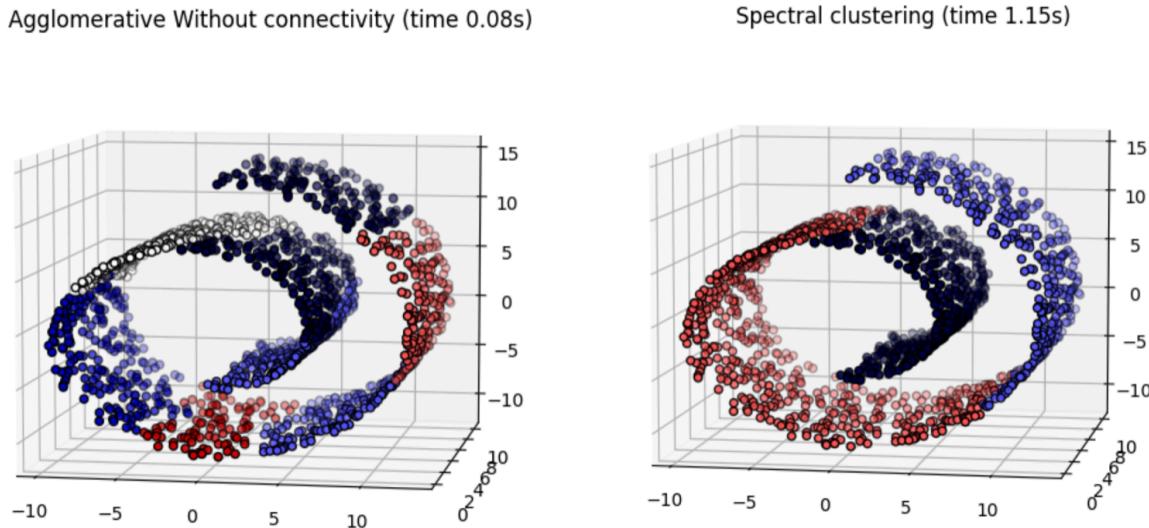


Fig. 7.5 Nube de puntos de movimiento suizo agrupada usando distancias euclidianas (izquierda) versus agrupada usando la teoría espectral (derecha).

Sin embargo, sabemos que **las distancias SP del camino más corto no son estrictamente respetuosas con la estructura del gráfico**. Por ejemplo, en un SBM con un cuello de botella importante, podemos tener dos nodos bastante cerca cuando están en la misma comunidad o en comunidades diferentes, pero este hecho no se refleja en la distancia SP.

Como resultado, hemos utilizado los tiempos de acierto y de desplazamiento en las lecciones prácticas de esta asignatura. Estos "tiempos" son en realidad distancias que cumplen con la geometría curva de los gráficos y pueden **aproximarse de manera eficiente mediante la teoría espectral**.

## 7.2.2. Tiempos de viaje y brecha

El tiempo de viaje  $CT(u, v) = H(u, v) + H(v, u)$  entre dos nodos  $u$  y  $v$  es el tiempo esperado que necesita una caminata aleatoria que parte de  $u$  para alcanzar  $v$  (tiempo de golpe  $H(u, v)$ ) y regresa (presiona aude  $H(v, u)$ ).

Como sabemos por las lecciones prácticas, los tiempos de viaje están relacionados con el **tamaño relativo del cuello de botella** (o constante de Cheeger) que se estima mediante la **brecha espectral**. Si el cuello de botella/brecha espectral es pequeño, la caminata aleatoria toma mucho tiempo para viajar de una comunidad a otra (ver [Fig. 7.6](#)).

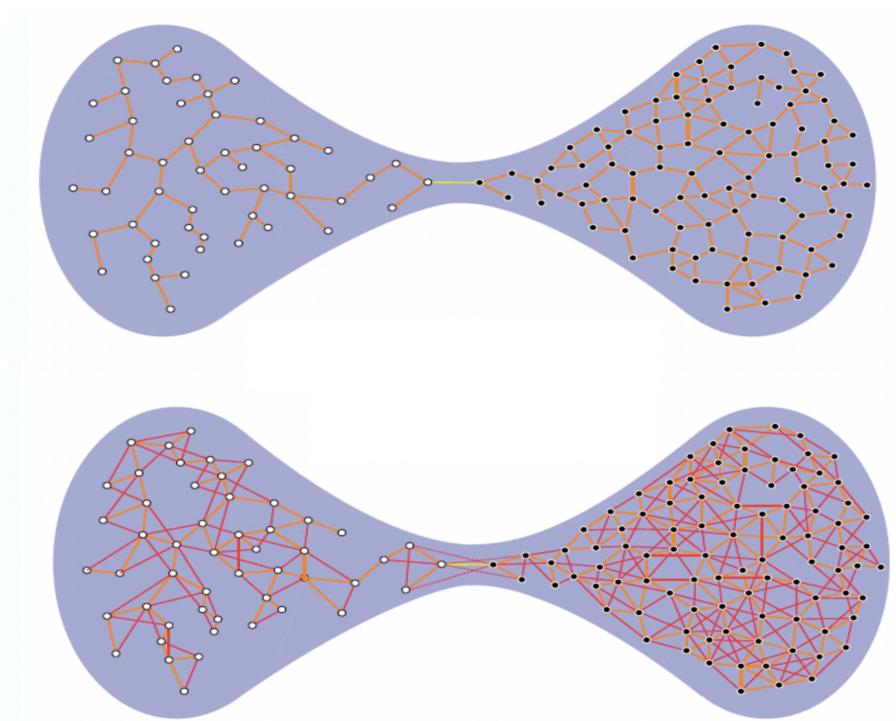
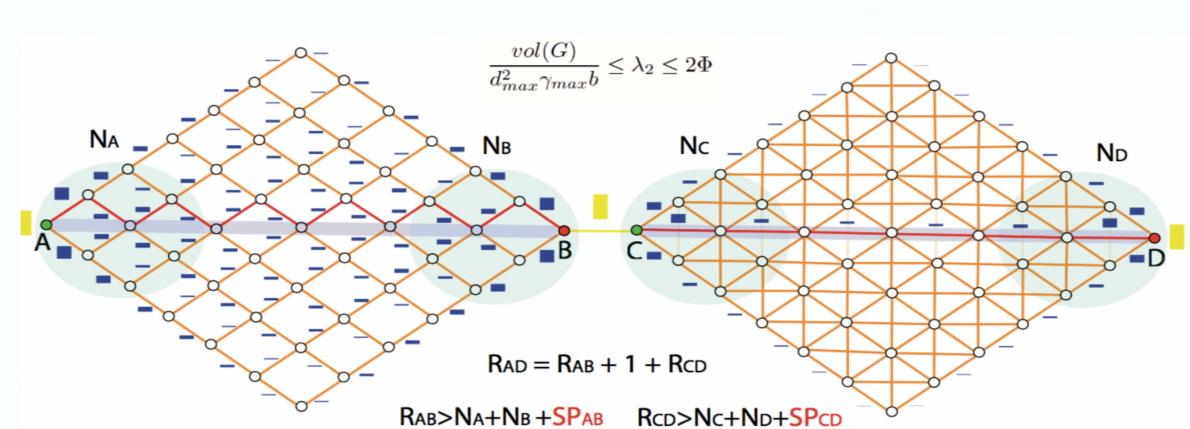


Fig. 7.6 Tiempos de viaje (CT) y brecha espectral. Arriba: una pequeña brecha espectral produce TC grandes. Abajo: una brecha espectral más grande produce CT más pequeños. Fuente: [Francisco Escolano en la Revista de Reconocimiento de Patrones](#)

**El Lovasz atado**. Como veremos a continuación, una pequeña brecha espectral es esencial para obtener TC informativos. En otras palabras, si la brecha espectral no es pequeña, el tiempo de viaje se puede aproximar de la siguiente manera

$$CT(u, v) \approx \frac{1}{d_u} + \frac{1}{d_v} .$$

Esta fue una observación clave de [von Luxburg et al.](#), basado en un límite propuesto por Lovasz. Este límite, ilustrado en [la Fig. 7.7](#), simplemente dice que: cuanto menor es la brecha espectral, más diferentes son el CT del  $\frac{1}{d_u} + \frac{1}{d_v}$ .



$$\left| \frac{1}{vol(G)} CT_{ij} - \left( \frac{1}{d_i} + \frac{1}{d_j} \right) \right| \leq 2 \left( \frac{1}{\lambda_2} + 2 \right) \frac{w_{max}}{d_{min}^2}$$

Fig. 7.7 Ilustración del Lovasz atado. Fuente: [Francisco Escolano en la Revista de Reconocimiento de Patrones](#)

A priori, ésta es una conclusión muy simple pero tiene implicaciones profundas. Los tiempos de viaje **no son informativos** en gráficos KNN grandes, gráficos Erdos-Renyi grandes, entre otros gráficos con una gran brecha espectral (cerca de 1) estudiado en el apartado anterior.

[Saltar al contenido principal](#)

Una forma de hacer que los CT sean informativos es volver a **cablear** el gráfico manteniendo la brecha espectral lo más pequeña posible. Por ejemplo, en [la Fig. 7.7](#) podemos ver que la comunidad de la derecha es **más densa** que la de la izquierda y esto reduce la brecha espectral.

### 7.2.3. Tiempos de viaje

Para la versión espectral de los tiempos de viaje, confiamos en la función de Green tal como se hace en el artículo todavía clásico de mi mentor [Edwin Hancock](#):

**La función de Green**. Una matriz clave para calcular los tiempos de impacto y de viaje es la llamada **función de Green**. Go pseudoíntverso  $\Delta^+$  del laplaciano  $\Delta$ . Para un gráfico conectado, tenemos la siguiente ecuación de componentes

$$G(u, v) = \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \phi_i(u) \phi_i(v)$$

**Tiempo de golpe**.  $H(u, v)$  está espectralmente determinado por

$$\begin{aligned} H(u, v) &= \text{vol}(G) [G(v, v) - G(u, v)] \\ &= \text{vol}(G) \left[ \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \phi_i(v)^2 - \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \phi_i(u) \phi_i(v) \right] \\ &= \text{vol}(G) \sum_{i=2}^{|V|} \frac{1}{\lambda_i} [\phi_i(v)^2 - \phi_i(u) \phi_i(v)] \\ &= \text{vol}(G) \frac{1}{\lambda_2} [\phi_2(v)^2 - \phi_2(u) \phi_2(v)] + \dots + \text{vol}(G) \frac{1}{\lambda_n} [\phi_n(v)^2 - \phi_n(u) \phi_n(v)]. \end{aligned}$$

Desde  $\lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_n$ , la importancia de cada término en la suma es inversamente proporcional a la correspondiente  $\lambda_i$ . Esto significa que si  $\lambda_2 \rightarrow 0$  (brecha espectral muy pequeña), tenemos:

$$H(u, v) \approx \text{vol}(G) [\phi_2(v)^2 - \phi_2(u) \phi_2(v)].$$

**Recordatorio**. Es clave señalar que en general  $H(u, v) \neq H(v, u)$ .

**Tiempo de viaje**.  $CT(u, v) = H(u, v) + H(v, u)$  está espectralmente determinado por los cálculos espetrales de  $H(u, v)$  y  $H(v, u)$ :

$$\begin{aligned} \frac{CT(u, v)}{\text{vol}(G)} &= H(u, v) + H(v, u) \\ &= G(v, v) - G(u, v) + G(u, u) - G(v, u) \\ &= G(u, u) + G(v, v) - 2G(u, v). \end{aligned}$$

**Los tiempos de viaje son métricas**. Entonces,  $CT(u, v) = \text{vol}(G) [G(u, u) + G(v, v) - 2G(u, v)]$  porque la función de Green es comutativa  $G(u, v) = G(v, u)$ . Como resultado,  $CT(u, v)$  es formalmente una **distanzia** entre dos nodos.

Entonces, el cálculo espectral del tiempo de viaje se puede derivar fácilmente y da como resultado

$$CT(u, v) = \text{vol}(G) \sum_{i=2}^{|V|} \frac{1}{\lambda_i} [\phi_i(u) - \phi_i(v)]^2.$$

De nuevo, si  $\lambda_2 \rightarrow 0$ , tenemos:

$$CT(u, v) = \text{vol}(G) \frac{1}{\lambda_2} [\phi_2(u) - \phi_2(v)]^2.$$

No sorprende ver que el tiempo de viaje entre nodos en un gráfico cambia según la estructura del gráfico. Vea el ejercicio a continuación.

[Saltar al contenido principal](#)

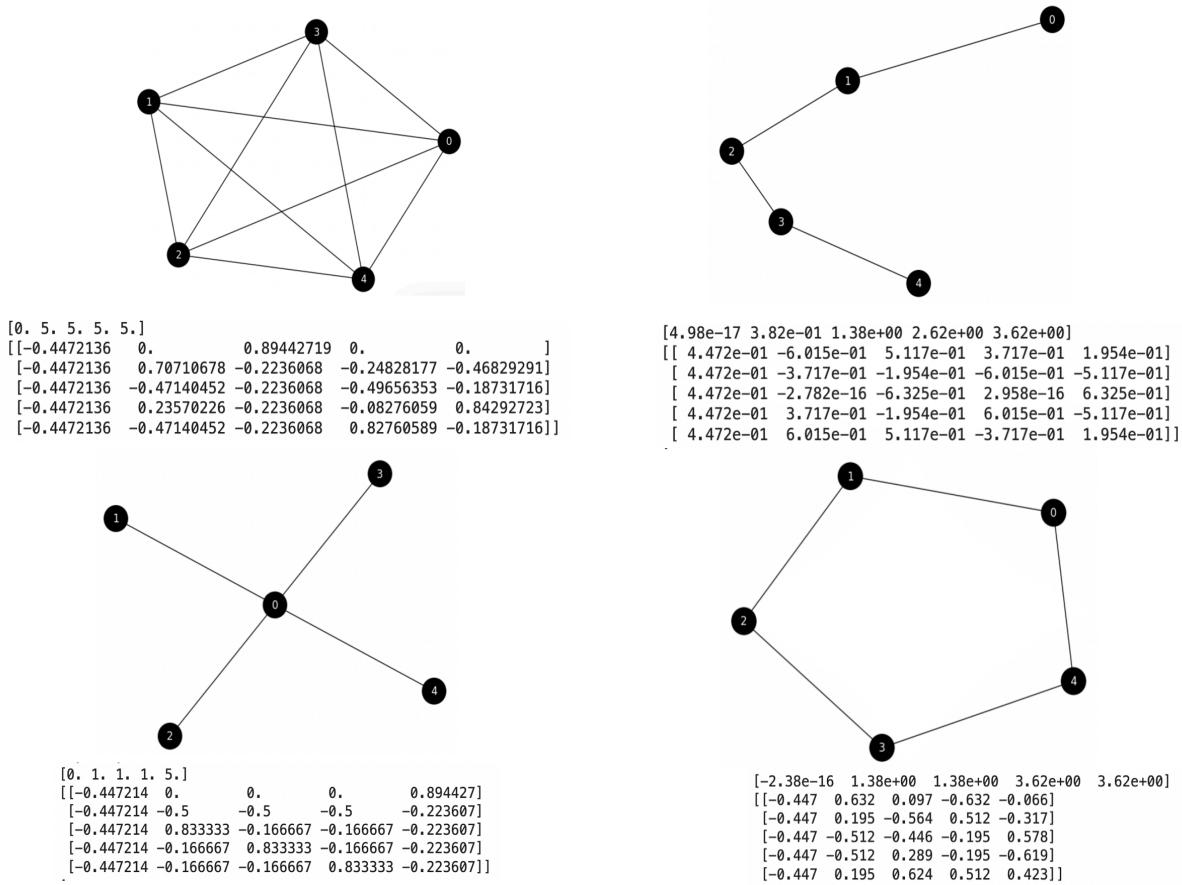


Fig. 7.8 Espectros y valores propios del laplaciano no normalizado de varios gráficos básicos.

**Ejercicio**. Considere los gráficos Fig. 7.8. Supongamos que solo tenemos la brecha espectral y el vector de Fiedler del laplaciano especificado en la figura.

Luego, aproxime los tiempos de viaje entre nodos 0 y 4 en los cuatro gráficos:  $K_5, S_5, P_5$  y  $C_5$  en la figura. Tenga en cuenta que **en lugar de utilizar los espectros y valores propios del laplaciano normalizado**, aquí utilizamos los del no normalizado. Usar únicamente  $\lambda_2$  y  $\phi_2$  y comentar el significado del resultado: ¿es **consistente** con la teoría? Cómo se relaciona con el valor aproximado  $CT(u, v) = \frac{1}{d_u} + \frac{1}{d_v}$ . Toma dos decimales. **Gráfico completo**. Tenemos que el volumen de esta gráfica es

$$\text{vol}(G) = 5 \cdot 4 = 20 \text{ y } \lambda_2 = 5. \text{ Entonces}$$

$$\begin{aligned} CT(0, 4) &= \text{vol}(G) \frac{1}{\lambda_2} [\phi_2(0) - \phi_2(4)]^2 \\ &= 20 \cdot \frac{1}{5} [\phi_2(0) - \phi_2(4)]^2 = 20 \cdot \frac{1}{5} [0 - (-0.47)]^2 \\ &= 20 \cdot \frac{1}{5} \cdot 0.22 = 0.88 \end{aligned}$$

**Gráfico estelar**. Tenemos que el volumen de esta gráfica es  $\text{vol}(G) = 4 + 4 = 8$  y  $\lambda_2 = 1$ . Entonces

$$\begin{aligned} CT(0, 4) &= \text{vol}(G) \frac{1}{\lambda_2} [\phi_2(0) - \phi_2(4)]^2 \\ &= 8 \cdot \frac{1}{1} [\phi_2(0) - \phi_2(4)]^2 = 8 \cdot \frac{1}{1} [0 - (-0.16)]^2 \\ &= 8 \cdot \frac{1}{1} \cdot 0.02 = 0.16 \end{aligned}$$

**Gráfico de ruta**. Tenemos que el volumen de esta gráfica es  $\text{vol}(G) = 1 + 2 + 2 + 2 + 1 = 8$  y  $\lambda_2 = 0.38$ . Entonces

[Saltar al contenido principal](#)

$$\begin{aligned}
 CT(0, 4) &= \text{vol}(G) \frac{1}{\lambda_2} [\phi_2(u) - \phi_2(v)]^2 \\
 &= 8 \cdot \frac{1}{0.38} [\phi_2(0) - \phi_2(4)]^2 = 8 \cdot \frac{1}{0.38} [0 - 0.6]^2 \\
 &= 8 \cdot \frac{1}{0.38} \cdot 0.36 = 7.57
 \end{aligned}$$

**Gráfico de ciclo**. Tenemos que el volumen de esta gráfica es  $\text{vol}(G) = 5 \cdot 2 = 10$  y  $\lambda_2 = 0.13$ . Entonces

$$\begin{aligned}
 CT(0, 4) &= \text{vol}(G) \frac{1}{\lambda_2} [\phi_2(u) - \phi_2(v)]^2 \\
 &= 10 \cdot \frac{1}{0.13} [\phi_2(0) - \phi_2(4)]^2 = 10 \cdot \frac{1}{0.13} [0.63 - 0.19]^2 \\
 &= 10 \cdot \frac{1}{0.13} \cdot 0.19 = 14.6
 \end{aligned}$$

**Comentarios**. Tenemos los siguientes CT entre nodos 0 y 4 en los diferentes gráficos:

$$K_5 : CT(0, 4) = 0.88$$

$$S_5 : CT(0, 4) = 0.16$$

$$P_5 : CT(0, 4) = 7.57$$

$$C_5 : CT(0, 4) = 14.6$$

**Los grafos completos** se caracterizan por tener el mismo CT entre cualquier par de nodos. Este valor es pequeño  $CT(0, 4) = 0.88$  porque los paseos aleatorios tienen una probabilidad uniforme  $\frac{1}{n-1}$  de llegar a cada uno de los nodos en un solo paso. La TC aproximada es

$$CT(0, 4) \approx \frac{1}{d_0} + \frac{1}{d_4} = \frac{1}{n-1} + \frac{1}{n-1} = 2 \frac{1}{n-1} = 2 \frac{1}{4} = 0.5.$$

**Los gráficos de estrellas** se caracterizan por tener el mismo CT entre el nodo central 0 y el resto (por ejemplo 4). Este valor es menor que el del gráfico completo.  $CT(0, 4) = 0.16$  porque los paseos aleatorios tienen una probabilidad uniforme  $\frac{1}{n-1}$  de llegar a cada uno de los nodos no centrales en un solo paso (como en el gráfico completo). Sin embargo, los nodos no centrales tienen una probabilidad unitaria de llegar al nodo central (recuerde que los tiempos de llegada son asimétricos en general). La TC aproximada es

$$CT(0, 4) \approx \frac{1}{d_0} + \frac{1}{d_4} = \frac{1}{n-1} + 1 = \frac{n-1+1}{n-1} = \frac{n}{n-1} = \frac{5}{4} = 1.25.$$

Esto es claramente una sobreestimación porque el valor de Fiedler para este gráfico es menor que el del gráfico completo. **Los gráficos de ruta** se caracterizan por tener CT grandes para nodos distantes. Esto se debe al hecho de que el gráfico de ruta es formalmente un árbol, es decir, no tiene bucles. Como resultado, el CT entre un par de nodos en este gráfico es **proporcional a la distancia del camino más corto (SP)** entre los nodos. En este caso, tenemos

$$CT(0, 4) = 7.57 \approx 2 \cdot 3 = 6. \text{ La TC aproximada es}$$

$$CT(0, 4) \approx \frac{1}{d_0} + \frac{1}{d_4} = 1 + 1 = 2.$$

Esto es claramente una subestimación porque el valor de Fiedler para este gráfico es menor que el del gráfico completo y el del gráfico en estrella. **Las gráficas de ciclo** se caracterizan por tener CT de gran tamaño para los nodos distantes. Observe que la probabilidad de salir de un nodo es siempre

1/2, pero podemos regresar con la misma probabilidad. Por lo tanto, para llegar al nodo 4 de 0, podemos pasar por los nodos 1, 2 y 3 o ir directamente a 4. Esto conduce al CT más grande en estos gráficos analizados:  $CT(0, 4) = 14.6$ . La TC aproximada es

$$CT(0, 4) \approx \frac{1}{d_0} + \frac{1}{d_4} = \frac{1}{2} + \frac{1}{2} = 1.$$

Esto es claramente una subestimación porque el valor de Fiedler para este gráfico está entre el del gráfico de estrellas y el del gráfico completo.

[Saltar al contenido principal](#)

## 7.2.4. Incrustación de tiempos de viaje

Como hemos visto antes, al estudiar dónde viven los datos, **incrutar** los nodos de un gráfico consiste en encontrar una representación vectorial de cada nodo  $\mathbf{z}_u$ ,  $u \in V$  donde la distancia euclídea  $\|\mathbf{z}_u - \mathbf{z}_v\|$  es consistente con la **distancia estructural** entre los nodos  $u$  y  $v$  en el gráfico.

En el artículo original de [Tenenbaum et al.](#), tenemos eso haciendo

$$\mathbf{z}_u = [\phi_2(u) \ \phi_3(u) \ \phi_4(u) \ \dots \ \phi_n(u)]^T$$

implica que  $\|\mathbf{z}_u - \mathbf{z}_v\|$  está **estadísticamente correlacionado** con las distancias de camino más cortas SP entre los nodos. Este hecho fue clave para encontrar correlaciones entre la información de los vectores propios y distancias estructurales más interesantes, como los tiempos de viaje.

Dado que los tiempos de viaje son, por definición, una distancia, es trivial comprobar que las columnas de esta matriz

$$\mathbf{Z} = \sqrt{\text{vol}(G)} \mathbf{\Lambda}^{-1/2} \mathbf{\Phi}^T \Rightarrow CT(u, v) = \|\mathbf{z}_u - \mathbf{z}_v\|^2.$$

La prueba es como sigue:

- Matriz  $\mathbf{Z}$  tiene dimensiones originales  $n \times n$  y la incrustación  $\mathbf{z}_u$  de nodo  $u$  está en la columna  $u$ .
- Veamos la estructura de una fila  $k$ , que codifica el  $k$ -ésima dimensión de la incrustación  $\mathbf{z}_u$ .

$$\mathbf{Z}_{k:} = \mathbf{\Lambda}^{-1/2} [\phi_1(u) \ \phi_2(u) \ \phi_3(u) \ \dots \ \phi_n(u)],$$

dónde  $\mathbf{\Phi}$  es la matriz de vectores propios (columnas desde el vector propio más pequeño hasta el más grande) y  $\mathbf{\Lambda} = \text{diag}(\lambda_1 \ \lambda_2 \ \dots \ \lambda_n)$  es la matriz diagonal con el espectro del laplaciano en la diagonal.

Luego, descartando el  $\sqrt{\text{vol}(G)}$  por un tiempo tenemos

$$\mathbf{Z}_{k:} = \left[ \frac{\phi_k(1)}{\sqrt{\lambda_k}} \ \frac{\phi_k(2)}{\sqrt{\lambda_k}} \ \frac{\phi_k(3)}{\sqrt{\lambda_k}} \ \dots \ \frac{\phi_k(n)}{\sqrt{\lambda_k}} \right].$$

- En realidad, la primera fila está normalizada por  $\lambda_1 = 0$  y asumimos que esta fila es:

$$\mathbf{Z}_{1:} = \left[ \frac{\phi_1(1)}{\sqrt{\lambda_1}} \ \frac{\phi_1(2)}{\sqrt{\lambda_1}} \ \frac{\phi_1(3)}{\sqrt{\lambda_1}} \ \dots \ \frac{\phi_1(n)}{\sqrt{\lambda_1}} \right] = [0 \ 0 \ 0 \ \dots \ 0].$$

- Hay **tantas filas como dimensiones de la incrustación** y la primera dimensión siempre tiene un valor cero. Esto significa que todas las incrustaciones están centradas.
- Consideremos, por ejemplo, que sólo tenemos la información del vector de Fiedler. Entonces, tenemos solo dos filas (la primera que es todo ceros y la segunda). Si descartamos la primera dimensión tenemos (después de incorporar el término  $\sqrt{\text{vol}(G)}$ ):

$$\mathbf{Z}_{2:} = \sqrt{\text{vol}(G)} \left[ \frac{\phi_2(1)}{\sqrt{\lambda_2}} \ \frac{\phi_2(2)}{\sqrt{\lambda_2}} \ \frac{\phi_2(3)}{\sqrt{\lambda_2}} \ \dots \ \frac{\phi_2(n)}{\sqrt{\lambda_2}} \right] = [\mathbf{z}_1 \ \mathbf{z}_2 \ \mathbf{z}_3 \ \dots \ \mathbf{z}_n] \dots$$

y obviamente incorporando

$$\|\mathbf{z}_u - \mathbf{z}_v\| = \text{vol}(G) \left[ \frac{\phi_2(u)}{\sqrt{\lambda_2}} - \frac{\phi_2(v)}{\sqrt{\lambda_2}} \right]^2 = \text{vol}(G) \frac{1}{\lambda_2} [\phi_2(u) - \phi_2(v)]^2 \approx CT(u, v).$$

y de manera similar para  $d$  dimensiones, donde el tiempo de viaje es exacto si  $d \rightarrow n$ . Más generalmente

[Saltar al contenido principal](#)

$$\|\mathbf{z}_u - \mathbf{z}_v\| = \text{vol}(G) \sum_{k=2}^d \left[ \frac{\phi_k(u)}{\sqrt{\lambda_k}} - \frac{\phi_k(v)}{\sqrt{\lambda_k}} \right]^2 = \text{vol}(G) \sum_{k=2}^d \frac{1}{\lambda_k} [\phi_k(u) - \phi_k(v)]^2 \approx CT(u, v).$$

¡La convergencia al número óptimo de dimensiones es más rápida cuando la brecha espectral es muy pequeña !

**Ejercicio** . Volvamos a los gráficos de [la figura 7.8](#) . Calcular las incorporaciones CT de nodos.0y4en cada gráfico. Considera dos dimensiones (además del cero) y tome dos decimales. Trazar el2Dde todos los nodos del plano y comentar los resultados.

**Gráfico completo** . Tenemos que el volumen de esta gráfica es

$$\text{vol}(G) = 5 \cdot 4 = 20 \text{ y } \lambda_2 = \lambda_3 = 5. \text{ Entonces}$$

$$\begin{aligned}\mathbf{Z}_{2:} &= \text{vol}(G) \begin{bmatrix} \frac{\phi_2(1)}{\sqrt{5}} & \frac{\phi_2(2)}{\sqrt{5}} & \dots & \frac{\phi_2(5)}{\sqrt{5}} \end{bmatrix} = \frac{20}{\sqrt{5}} \cdot [+0.00 + 0.70 - 0.47 + 0.23 - 0.47] \\ \mathbf{Z}_{3:} &= \text{vol}(G) \begin{bmatrix} \frac{\phi_3(1)}{\sqrt{5}} & \frac{\phi_3(2)}{\sqrt{5}} & \dots & \frac{\phi_3(5)}{\sqrt{5}} \end{bmatrix} = \frac{20}{\sqrt{5}} \cdot [+0.89 - 0.22 - 0.22 - 0.22 - 0.22]\end{aligned}$$

ver que el2Dincrustación de nodos3y5(2y4en [la figura 7.9](#) ) tienen la misma incrustación, es decir, chocan. **Gráfico estelar** . Tenemos que el volumen de esta gráfica es

$$\text{vol}(G) = 4 + 4 = 8 \text{ y } \lambda_2 = \lambda_3 = 1. \text{ Entonces}$$

$$\begin{aligned}\mathbf{Z}_{2:} &= \text{vol}(G) \begin{bmatrix} \frac{\phi_2(1)}{\sqrt{1}} & \frac{\phi_2(2)}{\sqrt{1}} & \dots & \frac{\phi_2(5)}{\sqrt{1}} \end{bmatrix} = \frac{8}{\sqrt{1}} \cdot [+0.00 - 0.50 + 0.83 - 0.16 - 0.16] \\ \mathbf{Z}_{3:} &= \text{vol}(G) \begin{bmatrix} \frac{\phi_3(1)}{\sqrt{1}} & \frac{\phi_3(2)}{\sqrt{1}} & \dots & \frac{\phi_3(5)}{\sqrt{1}} \end{bmatrix} = \frac{8}{\sqrt{5}} \cdot [+0.00 - 0.50 - 0.16 + 0.83 - 0.16]\end{aligned}$$

Mira eso en el2Dincrustación, nodo5(4en [la Fig. 7.9](#) ) está más cerca del nodo1(0en la figura) que otros. **Gráfico de ruta** . Tenemos que el volumen de esta gráfica es

$$\text{vol}(G) = 1 + 2 + 2 + 2 + 1 = 8 \text{ y } \lambda_2 = 0.38, \lambda_3 = 1.38. \text{ Entonces}$$

$$\begin{aligned}\mathbf{Z}_{2:} &= \text{vol}(G) \begin{bmatrix} \frac{\phi_2(1)}{\sqrt{0.38}} & \frac{\phi_2(2)}{\sqrt{0.38}} & \dots & \frac{\phi_2(5)}{\sqrt{0.38}} \end{bmatrix} = \frac{8}{\sqrt{0.38}} \cdot [-0.60 - 0.37 - 0.27 + 0.37 + 0.60] \\ \mathbf{Z}_{3:} &= \text{vol}(G) \begin{bmatrix} \frac{\phi_3(1)}{\sqrt{1.38}} & \frac{\phi_3(2)}{\sqrt{1.38}} & \dots & \frac{\phi_3(5)}{\sqrt{1.38}} \end{bmatrix} = \frac{8}{\sqrt{1.38}} \cdot [+0.51 - 0.19 - 0.63 - 0.19 + 0.51]\end{aligned}$$

ver que el2DLa incrustación en [la Fig. 7.9](#) es incluso más simétrica que la proporcionada por el “diseño de resorte” de Networkx. Esto se debe a que los valores propios son diferentes y añaden más flexibilidad al dibujo. Sin embargo, lo esencial aquí es que los vectores propios capturen bien las simetrías del gráfico.

**Gráfico de ciclo** . Tenemos que el volumen de esta gráfica es

$$\text{vol}(G) = 5 \cdot 2 = 10 \text{ y } \lambda_2 = \lambda_3 = 1.38. \text{ Entonces}$$

$$\begin{aligned}\mathbf{Z}_{2:} &= \text{vol}(G) \begin{bmatrix} \frac{\phi_2(1)}{\sqrt{1.38}} & \frac{\phi_2(2)}{\sqrt{1.38}} & \dots & \frac{\phi_2(5)}{\sqrt{1.38}} \end{bmatrix} = \frac{10}{\sqrt{1.38}} \cdot [+0.63 + 0.19 - 0.51 - 0.51 - 0.19] \\ \mathbf{Z}_{3:} &= \text{vol}(G) \begin{bmatrix} \frac{\phi_3(1)}{\sqrt{1.38}} & \frac{\phi_3(2)}{\sqrt{1.38}} & \dots & \frac{\phi_3(5)}{\sqrt{1.38}} \end{bmatrix} = \frac{10}{\sqrt{1.38}} \cdot [+0.09 - 0.56 - 0.44 + 0.28 + 0.62]\end{aligned}$$

ver que el2DLa incrustación en [la Fig. 7.9](#) es bastante similar a la proporcionada por el “diseño de resorte” de Networkx, incluso cuando el segundo y tercer valor propio son similares. Esto se debe a que los vectores propios capturan bien las simetrías en el gráfico.

[Saltar al contenido principal](#)

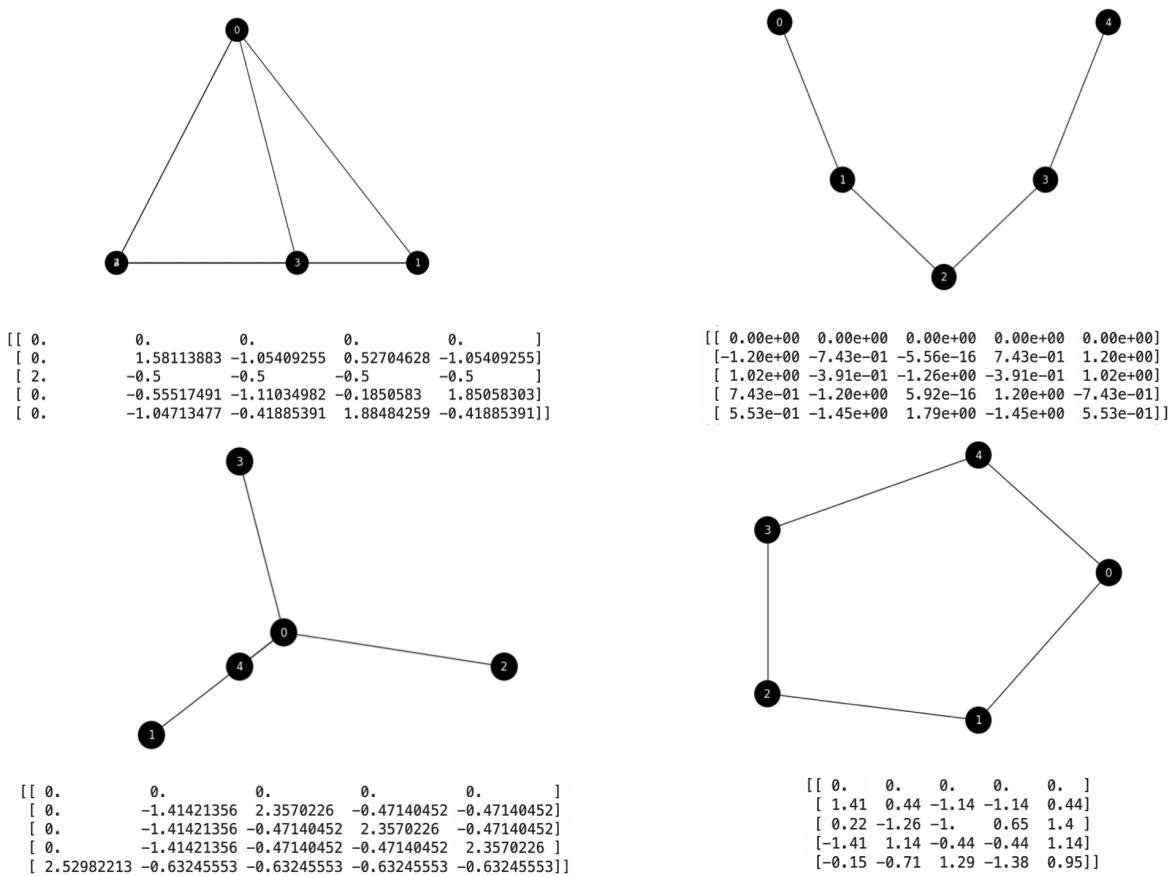


Fig. 7.9 Reconstrucción de los gráficos básicos a partir de sus incrustaciones de tiempos de viaje.

**Soluciones**. Las incorporaciones resultantes de Commute-Times se trazan en la Fig. 7.9, donde también trazamos los gráficos tomados de la segunda y tercera dimensiones de cada nodo. Tenga en cuenta que para el **gráfico completo**, los nodos 2 y 4 chocan. Para el nodo **del gráfico estelar** 4 está fuera de lugar. Sin embargo, para el **gráfico de ruta** y el **gráfico de ciclo**, el gráfico reconstruido se ajusta perfectamente al "diseño de resorte" de Networkx. Estos buenos resultados para los gráficos de ruta y ciclo **no tienen nodos notables** con grados más altos que otros. Los nodos con gran grado tienden a distorsionar la incrustación incluso cuando el gráfico tiene simetrías estructurales.

Solo para finalizar este tema con un vínculo con datos reales y las prácticas, trazamos las incorporaciones de tiempos de viaje del conocido **conjunto de datos Cora** en la Fig. 7.10. La parte superior derecha es la incorporación del segundo y tercer vector propio y es apenas informativa. Sin embargo, la parte inferior derecha incluye el cuarto vector propio y una proyección más avanzada (TSNE).

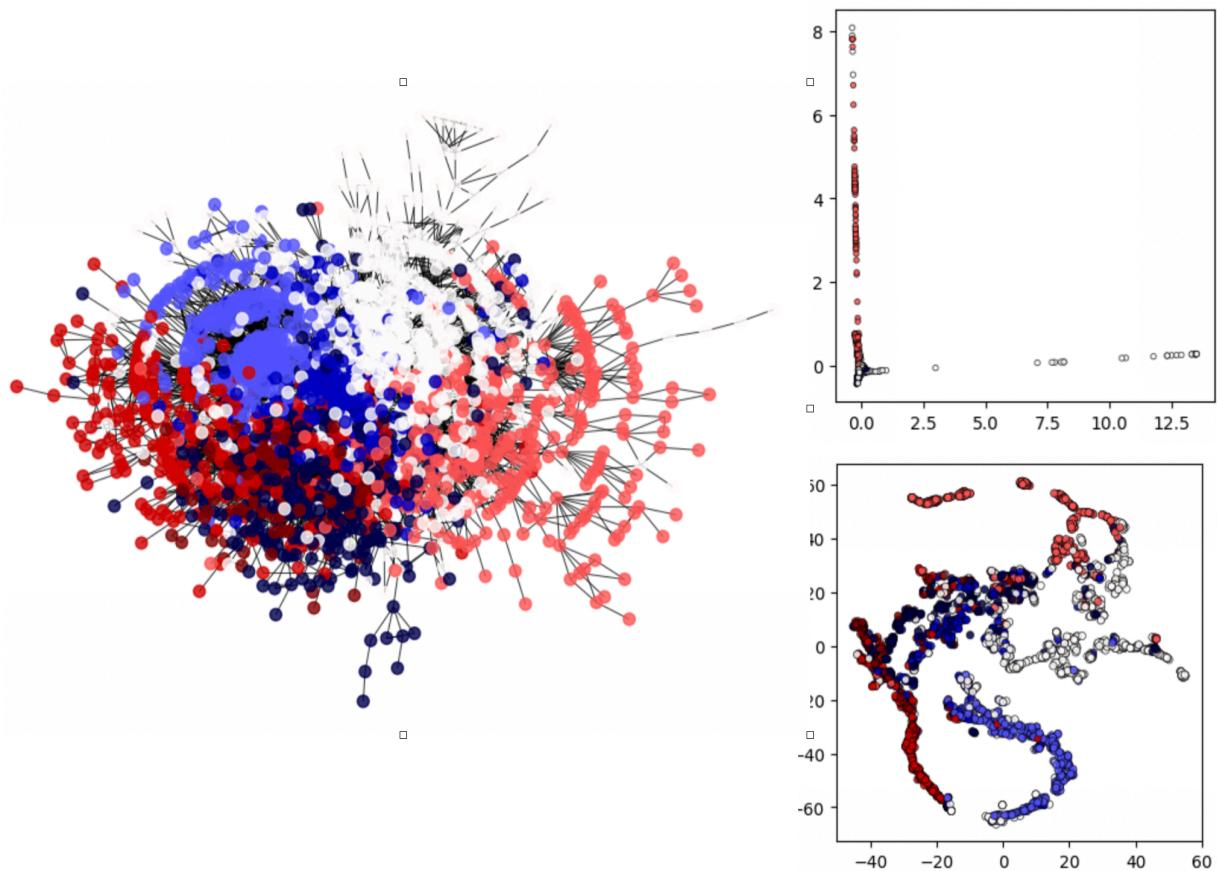


Fig. 7.10 Conjunto de datos de Cora: incorporaciones de tiempos de viaje.

< Anterior  
[6. Clasificación y particiones](#)

Próximo >  
[8. Introducción a la parte práctica de MD2024](#)