

Diseño de grafos pesados con n -ciclos persistentes

Omar Radhames Urquídez Calvo ¹
María Alejandra Valdez Cabrera ²

Resumen

Se presenta la resolución de un ejercicio del curso “Applications of topology to neuroscience”, impartido durante la 3ra Escuela de Análisis Topológico de Datos y Topología Estocástica por Carina Curto y Vladimir Itskov. En dicho ejercicio se utilizó el código de Matlab proporcionado por los instructores para diseñar grafos que, bajo una filtración basada en complejos de clique, tuvieran n -ciclos persistentes.

1 Introducción

Durante el curso de “Applications of topology to neuroscience” se vio el concepto de complejo de clique de un grafo no dirigido y cómo de esto surge una filtración natural al considerar grafos pesados.

Recuérdese que teniendo un grafo no dirigido, se le llama clique a cualquier subgrafo completo, es decir, a cualquier subgrafo en el cual cualquier par de vértices esten conectados entre sí. De esta forma se define el complejo clique de un grafo G no dirigido como un complejo simplicial abstracto que consta de los conjuntos de vértices en todos los cliques del grafo, denotado por $X(G)$. Por ejemplo, el complejo clique del grafo que se muestra en la Figura 1 es

$$\begin{aligned} X(G) = \{ \emptyset, [A], [B], [C], [D], [E], [F], [G], [H], [I], [A, B], [A, C], [B, C], \\ [B, E], [B, H], [B, I], [C, D], [D, E], [D, F], [E, H], [E, I], \\ [F, G], [H, I], [A, B, C], [B, E, I], [B, E, H], [B, I, H], \\ [E, I, H], [B, E, I, H] \}. \end{aligned}$$

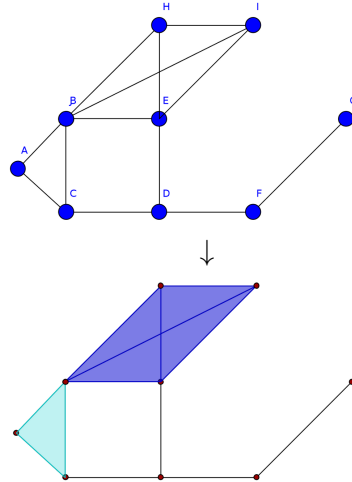


Figura 1: Grafo no dirigido (arriba) y su complejo clique correspondiente (abajo)

No es inusual trabajar con grafos pesados, en los cuales el peso de una arista corresponde al grado de correlación que existe entre los dos nodos conectados por dicha arista. Cuando se tiene un grafo G de este tipo, se puede considerar una secuencia de grafos $\{G_i\}$ que se obtiene de empezar solamente con los vértices de G y en cada paso agregar la arista de mayor peso que aún no haya sido agregada previamente.

De esta manera se obtiene una filtración

$$X(G_0) \subseteq X(G_1) \subseteq \dots \subseteq X(G_l) = X(G).$$

Durante el curso se proporcionó un código que, dado un grafo pesado, calcula los números de Betti de cada complejo simplicial en la filtración antes descrita y muestra una gráfica en la que se compara la cantidad de ciclos que tiene el grafo contra la cantidad de aristas que se le han agregado. Se propuso como ejercicio utilizar el código para encontrar grafos que tuvieran n -ciclos persistentes en la filtración. A continuación se muestran los grafos encontrados por nosotros para este ejercicio.

2 Grafos con 1-ciclos persistentes

Se empezó el ejercicio por encontrar un grafo que tuviera un 1-ciclo persistente. El primer intento fue un grafo muy sencillo de cinco nodos.

Puesto que se busca un 1-ciclo persistente, se optó por formar el polígono más grande posible (en este caso un pentágono) y luego trazar todas las aristas que se pudieran sin cerrar el ciclo. Se siguió el proceso que se muestra en la Figura 2.

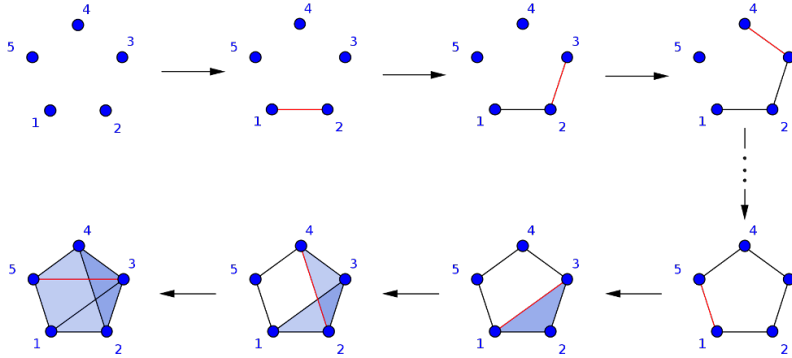


Figura 2: Filtración de complejos clique. En rojo se resalta en cada paso la arista nueva

Nótese que el 1-ciclo aparece cuando se agrega la quinta arista y se cierra finalmente cuando se agrega la octava. Así pues, nuestro ciclo en este caso tiene una persistencia de 3. El grafo pesado que da lugar a esta filtración se muestra en la Figura 3. Además en la misma figura se presenta la gráfica obtenida con el código proporcionado. Se observa que en efecto hay un 1-ciclo persistente.

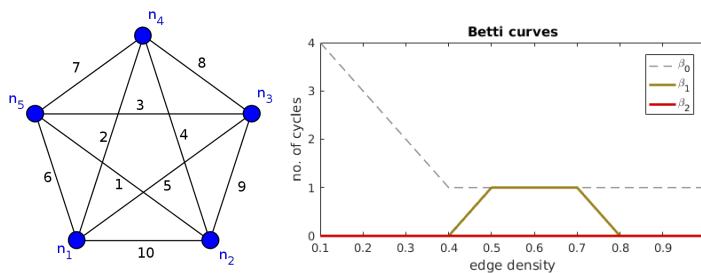


Figura 3: Primer grafo pesado (izquierda) y su gráfica de números de Betti correspondiente (derecha)

Siguiendo un procedimiento equivalente, se hizo otro grafo de 8 nodos. En la Figura 4 se muestra la matriz de adyacencia del grafo obtenido de esta manera y la gráfica de los números de Betti contra la

cantidad de aristas en el grafo. Obsérvese que además de tener un 1-ciclo persistente, aparece además un 2-ciclo que persiste durante poco tiempo y un 3-ciclo que nace en un complejo y muere en el paso siguiente.

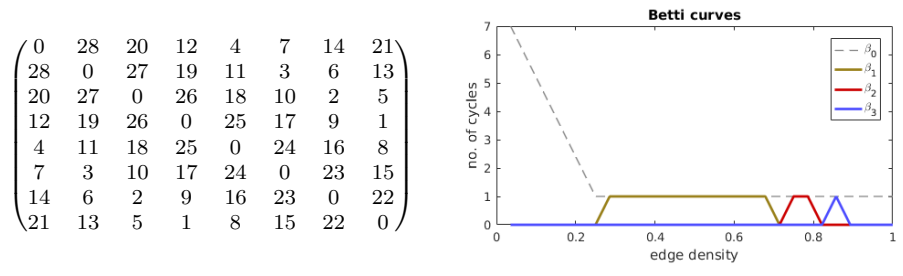


Figura 4: Matriz de Adyacencia de una matriz pesada de 8 nodos (izquierda) y su gráfica de números de Betti correspondiente

3 Grafos con 2-ciclos persistentes

Se procederá ahora para encontrar grafos que tengan 2-ciclos persistentes. Para esto se empezó por hacer un grafo cuyos nodos fueran los vértices de un cubo y se rellenan las 6 caras de dicho cubo, para crear un vacío, que es un 2-ciclo, dentro del cubo. Se procede triangular las caras de manera regular, como se muestra en la Figura 5.

Después de triángular las 6 caras, se traza la diagonal restante en cada una, para prolongar la persistencia del 2-ciclo formado. Finalmente se agregan las 3 diagonales del cubo. Nótese en el dibujo que el 2-ciclo nace cuando todas las caras están trianguladas, que es al agregar la arista número 18 y que al trazar la primera diagonal del cubo muere. En la Figura 6 se muestra la matriz de adyacencia del grafo pesado del que surge la filtración anterior y su correspondiente gráfica de números de Betti contra el número de aristas.

4 Grafos con 3-ciclos persistentes

Siguiendo la intuición de que el número de Betti-1 nos describe la cantidad de 1-ciclos (huecos) y Betti-2 el número de 2-ciclos (vacíos) se puede pensar en el número de Betti-3 como un “hipervacío”. Para ello se aprovechó que la estructura más familiar para el trabajo en cuatro

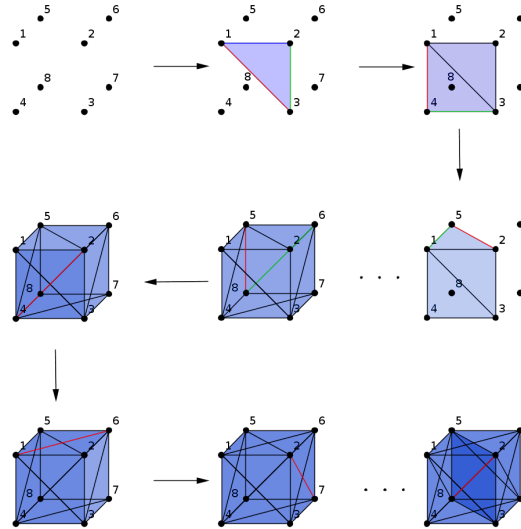


Figura 5: Filtración de complejos clique. En cada paso se resalta en verde y rojo las dos aristas más nuevas.

dimensiones sea precisamente el hipercubo. Se introducirá para ello el grafo hipercubo.

El grafo hipercubo Q_n es un grafo regular con 2^n vértices, que corresponden a los subconjuntos de un conjunto de n elementos. Dos vértices etiquetados por subconjuntos W y B están unidos por una arista si y sólo si W puede ser obtenido desde B añadiéndosele o quitándosele a este último un único elemento.

El nombre proviene del hecho de que un grafo hipercubo es un esqueleto unidimensional de un hipercubo geométrico. Así pues, Q_4 es la representación que deseamos y podemos apoyarnos en la imagen mostrada en la Figura 7.

Para representarla en la computadora se realizó el siguiente procedimiento: se numeraron los nodos externos con los números impares, del 1 al 15; mientras que los internos desde el 2 hasta el 16. Se puede ver que se cumple la siguiente relación:

- Si el vértice v es impar, entonces estará conectado a los nodos $(v-2), (v+2), (v-1)$ y $(v+3)$, todos en módulo 16, por supuesto.
- Si el vértice u es par, entonces estará conectado a los nodos $(u-6), (u+6), (u+1)$ y $(u-3)$, también en módulo 16.

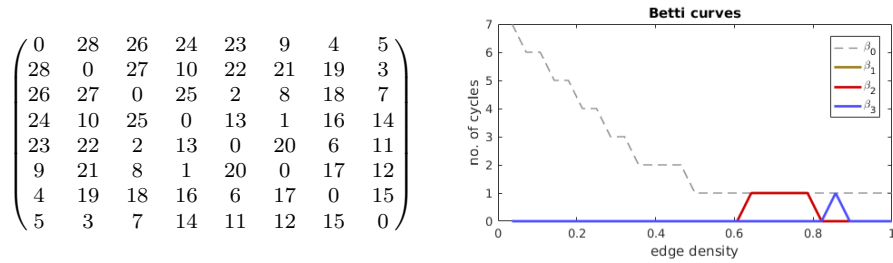


Figura 6: Matriz de Adyacencia de una matriz pesada de 8 nodos (izquierda) y su gráfica de números de Betti correspondiente

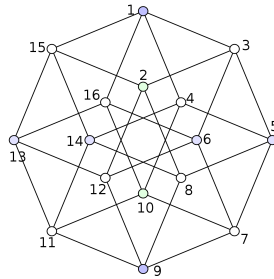


Figura 7: Grafo Hipercubo

Se sabe que para crear un vacío en un cubo lo primero que debemos hacer es completar sus seis caras: siguiendo esta intuición, se procedió al llenado de los 8 cubos que conforman a Q_4 . Para ilustrar esto obsérvese la Figura 8.

Los cuatro vértices que conforman cada cuadrado rojo forman un cubo con los otros cuatro vértices de cualquier otro cuadrado del mismo color. Lo mismo ocurre con los amarillos. Así pues, tras llenar los ocho cubos de la misma forma que en la sección anterior, de manera simultánea para cada uno de ellos, se obtiene el grafo pesado descrito por la matriz de adyacencia en la Figura 9, junto con la correspondiente gráfica de sus números de Betti contra la cantidad de aristas.

5 Apéndice

```
% script to compute and plot clique topology for small matrices,
% to be used during practicum time in Mexico TDA lectures
% calls: CliqueTop package and generate_sq_dist_mtx.m
```

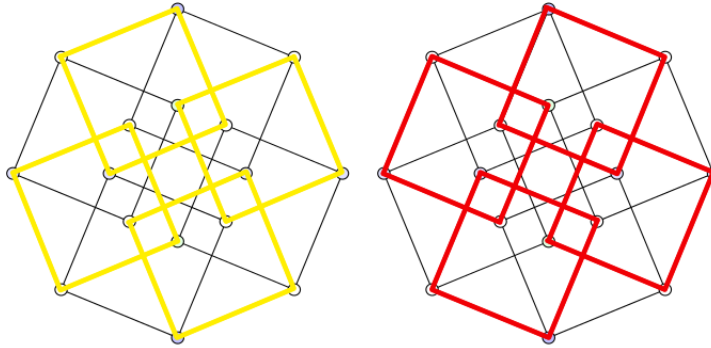


Figura 8: Grafo Hipercubo con cuadros importantes señalados

0	126	43	0	91	44	125	92	0	38	127	0	90	128	37	93
126	0	94	123	122	95	96	68	51	0	0	0	67	0	0	124
43	94	0	119	39	89	88	40	118	0	0	0	120	0	0	121
0	123	119	0	0	117	116	0	0	0	0	0	0	0	0	0
91	122	39	0	0	114	70	87	85	64	113	0	0	46	115	74
44	95	89	117	114	0	86	112	111	0	0	0	56	0	0	52
125	96	88	116	70	86	0	110	57	0	0	0	109	0	0	66
92	68	40	0	87	112	110	0	84	108	107	0	75	49	42	0
0	51	118	0	85	111	57	84	0	106	62	0	79	59	105	80
38	0	0	0	64	0	0	108	106	0	83	104	103	78	81	41
127	0	0	0	113	0	0	107	62	83	0	102	50	73	82	47
0	0	0	0	0	0	0	0	0	104	102	0	0	101	100	0
90	67	120	0	0	56	109	75	79	103	50	0	0	99	0	71
128	0	0	0	46	0	0	49	59	78	73	101	99	0	77	98
37	0	0	0	115	0	0	42	105	81	82	100	0	77	0	97
93	124	121	0	74	52	66	0	80	41	47	0	71	98	97	0

Figura 9: Matriz de Adyacencia del hipercubo.

```

% FIRST, create or load an nxn matrix .....
n = 25; d = 10;
%A = generate_sq_dist_mtx(n,d);
% A = rand(n);
%A = [0 1 2 3 4 5; 0 0 6 7 8 9; 0 0 0 10 11 12; 0 0 0 0 13 14;...
%     0 0 0 0 0 15; 0 0 0 0 0 0];

% 1-ciclos persistentes

%Pentagono
% 1 2 3 4 5
%A=[ 0,10, 5, 2, 6;... %1
%   0, 0, 9, 4, 1;... %2
%   0, 0, 0, 8, 3;... %3
%   0, 0, 0, 0, 7;... %4
%   0, 0, 0, 0, 0]; %5

% 1 2 3 4 5 6 7 8
    
```

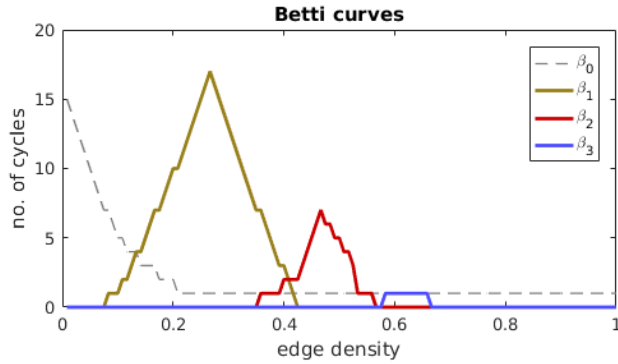


Figura 10: Gráfica de números de Betti correspondiente a la matriz en la Figura 9.

```
%A=[ 0,28,20,12, 4, 7,14,21;... %1
% 0, 0,27,19,11, 3, 6,13;... %2
% 0, 0, 0,26,18,10, 2, 5;... %3
% 0, 0, 0, 0,25,17, 9, 1;... %4
% 0, 0, 0, 0, 0,24,16, 8;... %5
% 0, 0, 0, 0, 0, 0,23,15;... %6
% 0, 0, 0, 0, 0, 0, 0,22;... %7
% 0, 0, 0, 0, 0, 0, 0, 0] %8
```

% 2-ciclos persistentes

```
%1 2 3 4 5 6 7 8
%A=[0,32, 0,30,29,26,18,14;... %1
% 0, 0,31,10,28,25,17,13;... %2
% 0, 0, 0,21,27,24,16,12;... %3
% 0, 0, 0, 0,23,22,15,11;... %4
% 0, 0, 0, 0, 0, 0, 0,20;... %5
% 0, 0, 0, 0, 0, 0,19, 0;... %6
% 0, 0, 0, 0, 0, 0, 0, 0;... %7
% 0, 0, 0, 0, 0, 0, 0, 0]; %8
```

```
% 1 2 3 4 5 6 7 8
%A= [ 0,28,26,24,23, 9, 4, 5;... %1
% 0, 0,27,10,22,21,19, 3;... %2
% 0, 0, 0,25, 2, 8,18, 7;... %3
% 0, 0, 0, 0,13, 1,16,14;... %4
% 0, 0, 0, 0, 0,20, 6,11;... %5
% 0, 0, 0, 0, 0, 0,17,12;... %6
% 0, 0, 0, 0, 0, 0, 0,15;... %7
%0, 0, 0, 0, 0, 0, 0, 0] %8
```



```
%3-ciclos persistentes
```

```
A= [0,126, 43, 0, 91, 44,125, 92, 0, 38,127, 0, 90,128, 37, 93; ...
    0, 0, 94,123,122, 95, 96, 68, 51, 0, 0, 0, 67, 0, 0,124; ...
    0, 0, 0,119, 39, 89, 88, 40,118, 0, 0, 0,120, 0, 0,121; ...
    0, 0, 0, 0, 0,117,116, 0, 0, 0, 0, 0, 0, 0, 0, 0; ...
    0, 0, 0, 0, 0,114, 70, 87, 85, 64,113, 0, 0, 46,115, 74; ...
    0, 0, 0, 0, 0, 0, 86,112,111, 0, 0, 0, 56, 0, 0, 52; ...
    0, 0, 0, 0, 0, 0, 0,110, 57, 0, 0, 0,109, 0, 0, 66; ...
    0, 0, 0, 0, 0, 0, 0, 0, 84,108,107, 0, 75, 49, 42, 0; ...
    0, 0, 0, 0, 0, 0, 0, 0, 0,106, 62, 0, 79, 59,105, 80; ...
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 83,104,103, 78, 81, 41; ...
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,102, 50, 73, 82, 47; ...
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,101,100, 0; ...
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 60, 99, 0, 71; ...
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 77, 98; ...
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 97; ...
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
% symmetrize and put zeros on diagonal to get M
```

```
M = (A + A') - 2*diag(diag(A));
```

```
M
```

```
% SECOND, compute clique topology of M - including betti 0 .....
```

```
rho = 1; % max edge density in (0,1]
```

```
tic
```

```
[bettis, edgeDensities]=compute_clique_topology(M,'MaxEdgeDensity',rho,
... 'ComputeBetti0', true);
```

```
toc
```

```
% third, plot the Betti curves for b_0,...,b_3 .....
```

```
colors = [.5 .5 .5; .6 .5 .1; .8 0 0; .3 .3 1];
```

```
figure
```

```
% plot matrix
```

```
subplot(3,3,1); imagesc(M); title('M');
```

```
% plot b_0 with a gray dashed line
```

```
subplot(3,1,2:3);
```

```
plot(edgeDensities,bettis(:,1),'--','color',colors(1,:), 'LineWidth',1);
```

```
hold on;
```

```
% now plot b_1,b_2,b_3
```

```
for i=2:size(bettis,2)
```

```
    plot(edgeDensities,bettis(:,i),'color',colors(i,:), 'LineWidth',2);
```

```
end;
```

```
xlabel('edge density')
```

```
ylabel('no. of cycles')
```

```
legend('\beta_0', '\beta_1', '\beta_2', '\beta_3')
```

```
title('Betti curves')
```

```

hold off;

% symmetrize and put zeros on diagonal to get M
M = (A + A') - 2*diag(diag(A));

M
% SECOND, compute clique topology of M - including betti 0 .....
rho = 1; % max edge density in (0,1]
tic
[bettis, edgeDensities]=compute_clique_topology(M,'MaxEdgeDensity',rho,
... 'ComputeBetti0', true);
toc

% third, plot the Betti curves for b_0,...,b_3 .....
colors = [.5 .5 .5; .6 .5 .1; .8 0 0; .3 .3 1];
figure
% plot matrix
subplot(3,3,1); imagesc(M); title('M');
% plot b_0 with a gray dashed line
subplot(3,1,2:3);
plot(edgeDensities,bettis(:,1),'--','color',colors(1,:),'LineWidth',1);
hold on;
% now plot b_1,b_2,b_3
for i=2:size(bettis,2)
    plot(edgeDensities,bettis(:,i),'color',colors(i,:),'LineWidth',2);
end;
xlabel('edge density')
ylabel('no. of cycles')
legend('\beta_0', '\beta_1', '\beta_2', '\beta_3')
title('Betti curves')
hold off;

```

Omar Radhames Urquidez Calvo
Departamento de Matemáticas,
 Universidad de Guanajuato,
 Camino del Jalisco 37A,
 Valenciana, Guanajuato, Gto.,
 omar.urquidez@cimat.mx

María Alejandra Valdez Cabrera
Departamento de Matemáticas,
 Universidad de Guanajuato,
 Camino del Jalisco 37A,
 Valenciana, Guanajuato, Gto.,
 maria.valdez@cimat.mx

Referencias

- [1] Curto, C. , “*What can Topology tell us about the Neural Code?*”
 BULLETIN OF THE AMERICAN MATHEMATICAL SOCIETY
 Volume 54, Number 1, January 2017, Pages 6378