Proyecto ATD: ¿Cómo calular números de Betti en R?

Rafael José González De Gouveia

June 11, 2016

Maestría en Probabilidad y Estadística. CIMAT

Objetivo

Reproducir los algoritmos para calcular los números de Betti en $\mathbb{R}^3$ en el lenguaje de programación R. Presentar los resultados en un formato interactivo que pueda ser utilizado sin conocimiento previo de R.

Resumen

En este proyecto se desarrolla un algoritmo para calcular los números de Betti a partir de una filtración del complejo de Vietoris-Rips para un radio fijo. Se utiliza la matriz de Frontera y la forma normal de Smith para calcular los números de Betti. La bibliografía utilizada son las Notas de Espinosa L. Malors disponibles en la página de ATD de CIMAT y el libro "Computational Topology" de H. Edelsbrunner y J. Harer. Posteriormente se utilizan estos algoritmos en conjunto con el paquete TDA en R para generar una aplicación interactiva que permita a un usuario, sin necesidad de conocer R, visualizar la evolución de la homología en un complejo simplicial a medida que varía el radio de la filtración de Vietoris-Rips. Esta aplicación se creó utilizando el paquete "Shiny" en Rstudio.

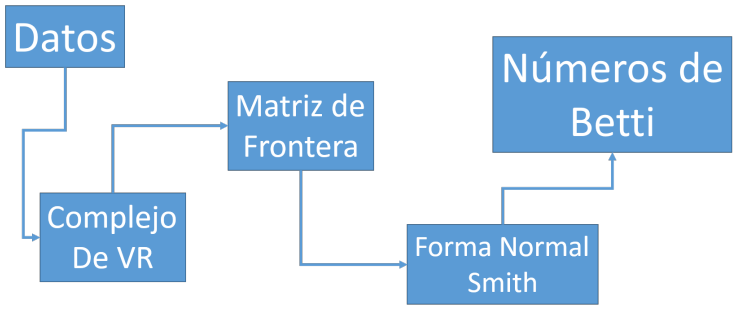
Contenido

1. Introducción.
2. Complejo de VR.
3. Reducción Matricial.
4. Shiny.
5. Ejemplo de uso.
6. Conclusiones.

Introducción

Los números de Betti son la dimensión de los grupos de homología y resumen las características topológicas. Caracterizan completamente la topología de las variedades de dos dimensiones. Para una variedad con d dimensiones tenemos $d+1$ números de Betti, β_p, con $0 \leq p < d$. El p-ésimo número de Betti indica un hoyo p-dimensional. En términos prácticos, los números de Betti cuentan las características topológicas, en una superficie sobre un espacio tridimensional, β_0 cuenta las componentes conexas, β_1 cuenta los túneles, β_2 cuenta los agujeros cerrados y los demás números de Betti son cero.

Para los alcances de este proyecto, es de interés desarrollar algoritmos para calcular los números de Betti, considerando el siguiente diagrama.



Los datos vienen por lo general en una matriz $n \times 3$, donde cada fila representa un punto y cada columna las coordenadas x, y, z. Luego, a partir de un radio fijo r, se genera el complejo de Vietoris-Rips asociado a este radio r, en R, esto se guarda en el tipo de variable lista. En este sentido conviene pensar al grupo de las p-cadenas como un espacio vectorial. Cada simplejo es una base de dicho espacio, por lo tanto cualquier subcomplejo puede verse como una combinación lineal de simplejos en el mismo sentido que en álgebra lineal.

Una vez en este formato se genera la matriz de frontera, que contiene información sobre cuales $(p-1)-$simplejos son cara de cuales $p-$simplejos, en la próxima sección se hará más clara esta noción. Finalmente se transforma la matriz de frontera a la forma normal de Smith, la cual es una reducción matricial similar a la forma canónica de Gauss Jordan cuando resolvemos sistemas de ecuaciones lineales. En las próximas secciones se harán más precisas estas nociones y se presenta un ejemplo del cálculo de números de Betti.

Complejo de VR

En primer lugar es necesario pasar de la nube de datos al complejo simplicial. La nube de datos la llamaremos X, la cual es una matriz $n \times 3$, donde cada fila representa un dato, y cada columna las coordenadas x, y, z de los datos. De manera similar a la **sección 1.3.1**, dado un radio r el complejo de VR de una nube de datos X es

$$\mathrm{VR}(X, r) = \{\sigma \subset X : \ \mathrm{diam}(\sigma) \leq 2r\},$$

donde σ son subcomplejos, y $\mathrm{diam}(\sigma) = \sup_{x_i, x_j \in \sigma} \{d(x_i, x_j)\}$, con d un distancia cualquiera; en nuestro caso utilizamos la distancia euclideana. De esta manera vemos que para verificar que un subconjunto de X es un simplejo VR debe ocurrir que la mayor de las distancias entre los puntos sea menor a $2r$. Con esto ya tenemos información suficiente para generar los complejos a partir de la nube de datos y un radio r. Los algoritmos para la distancia euclideana y el diámetro son:

```
#Distacia euclideana
eu.dist <- function(x,y){sqrt(sum((x-y)^2))}

#Diametro
diam <- function(X){
  n <- nrow(X);  d <- 0
  for(i in 1:(n-1)){    for(j in (i+1):n){
      d <- max(d,eu.dist(X[i,],X[j,]))  }  }  return(d)}
```

Para la construcción del complejo $\mathrm{VR}(X, r)$ empezamos por los 1-simplejos. Tenemos que revisar cuales de los pares $\{x_i, x_j\}$ están en $\mathrm{VR}(X, r)$, esto se puede hacer mediante dos ciclos "for" que busquen entre todos los datos, almacenando la información en una lista de la manera siguiente:

```
C1.list <- function(r,X){
  C1 <- list()
  for(i in 1:(nrow(X)-1)){
    for(j in (i+1):nrow(X)){
      if(eu.dist(X[i,],X[j,])<2*r){
        Caux <- list(c(i,j))
        C1 <- c(C1,Caux)  }  }  }  return(C1)}
```

Note que la lista obtenida en este pase es el grupo de 1-cadenas del complejo. Se procede de manera similar para hallar los grupos de 2- y 3-cadenas del complejo.

Una vez hecho esto, podemos visualizar el simplejo con ayuda del paquete "rgl" y las siguientes funciones para dibujar el complejo sobre la nube de datos:

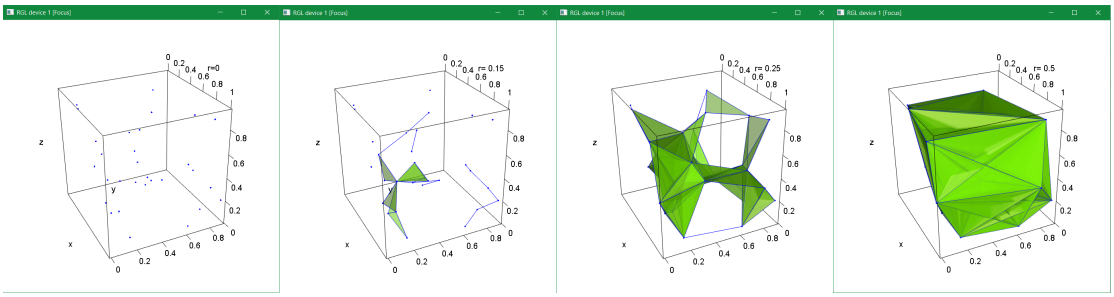
```
#pinta1 para graficar 1-simplejos, o aristas
pinta1 <- function(C,X){
  n <- length(C)
  for(k in 1:n){
    xyz <- X[C[[k]],]    #los puntos del k-esimo 1-simplejo
    lines3d(xyz,col=4)  }  }

#pinta2 para graficar 2.simplejos, o triangulos
pinta2 <- function(C,X,a=1){
  n <- length(C)
  for(k in 1:n){
    xyz <- X[C[[k]],]    #los puntos del k-esimo 2-simplejo
    triangles3d(xyz,col="lawngreen",alpha = a) #alpha es la transparencia  } }
```

Ejemplo

Tomemos 30 datos uniformes en el cubo $[0,1] \times [0,1] \times [0,1]$. Estos datos se pueden simular con:

```
n <- 30; X <- matrix(runif(3*n),nrow=n,ncol=3)
```

A continuación imágenes del complejo VR para diferentes radios, $r \in \{0, 0.15, 0.25, 0.5\}$:



Reducción matricial

Los grupos de homología de espacios triangulados pueden ser calculados de la matrices que representen el homomorfismo de frontera. Sus versiones reducidas proveen información de la dimensión del grupo de ciclos y del grupo de frontera, y con su diferencia se obtienen los números de Betti. A continuación se describe la matriz de frontera, con la cual podremos calcular los números de Betti luego de aplicarle una reducción matricial llamada forma normal de Smith.

Matriz de frontera

Para calcular la homología, combinamos la información de dos fuentes, una para los ciclos y otra que representa las fronteras. Sea K un complejo simplicial. Su matriz de frontera de orden p representa los $(p-1)$-simplejos como filas y los p-simplejos como columnas. Suponiendo un orden arbitrario pero fijo de los simplejos, para cada dimensión, la matriz se denota por $\partial_p = [a_i^j]$, donde $i = 1, \ldots, n_{p-1}$ (n_{p-1} es la cantidad de $(p-1)$-simplejos en K) y $j = 1, \ldots, n_p$ (n_p es la cantidad de p-simplejos en K), $a_i^j = 1$ si el i-ésimo

$(p-1)$-simplejo es cara del j-ésimo p-simplejo y $a_i^j = 0$ en otro caso. Dada una p-cadena, $c = \sum a_i \sigma_i$ la frontera puede ser calculada por un producto matricial,

$$\partial_p c = \begin{bmatrix} a_1^1 & a_1^2 & \cdots & a_1^{n_p} \\ a_2^1 & a_2^2 & \cdots & a_2^{n_p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_{p-1}}^1 & a_{n_{p-1}}^2 & \cdots & a_{n_{p-1}}^{n_p} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_p} \end{bmatrix}.$$

De manera coloquial, un conjunto de columnas representa una p-cadena y la suma de estas columnas es la frontera.

Operaciones de filas y columnas

Las filas de la matriz ∂_p forman una base del $(p-1)$-ésimo grupo de cadenas, C_{p-1}, y las columnas forman una base para el p-ésimo grupo de cadenas, C_p. Utilizamos dos tipos de operaciones de columnas para modificar la matriz sin cambiar su dimensión: (1) intercambiar las columnas k y l y (2) sumar la columna k con la columna l. Ambas pueden expresarse multiplicando por la derecha con una matriz $V = [v_i^j]$. Para intercambiar las columnas, tenemos que $v_k^l = v_l^k = 1$ y $v_i^i = 1$ para $i \neq k, l$, con el resto de entradas nulas. Para sumar la columna k a la l, tenemos que $v_k^l = 1$ y $v_i^i = 1$ para toda i, con el resto de entradas nulas, como se muestra en la Figura 1.

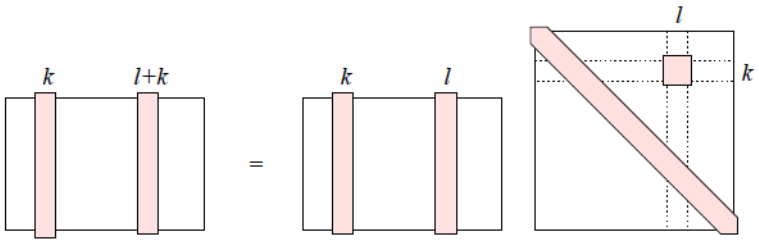


Figure 1: Operación por columna

De manera similar tenemos dos operaciones por fila, una para intercambiar dos filas y otra para sumar una fila a otra. Esto se traduce a un producto matricial por la izquierda con una matriz $U = [u_i^j]$. Para intercambiar dos filas, tenemos $u_k^l = u_l^k = 1$, $u_i^i = 1$ para $i \neq k, l$, con el resto de entradas nulas. Para sumar la k-ésima fila con la l-ésima fila tenemos $u_l^k = 1$, $u_i^i = 1$ para todo i, con el resto de entradas nulas, como en la figura 2. Las imágenes fueron extraídas del libro "Computational Topology" de H. Edelsbrunner y J. Harer.

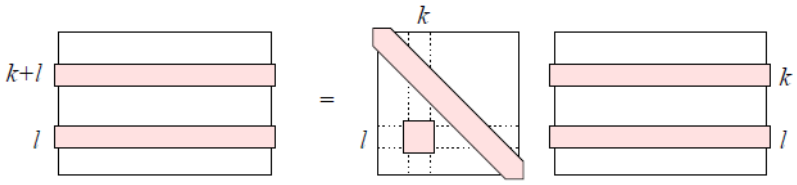


Figure 2: Operación por fila

Forma Normal de Smith

Utilizando operaciones de filas y columnas, podemos reducir la p-ésima matriz frontera a la Forma Normal de Smith. Para la aritmética en módulo 2, esto resulta en que un segmento de la diagonal es 1 y el resto es 0, como en la Figura 3.

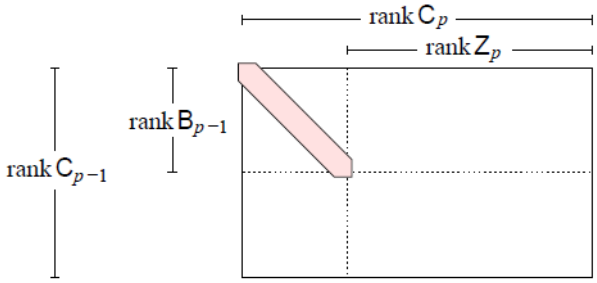


Figure 3: Forma Normal de Smith

Recuerde que n_p (la dimensión de C_p) es el número de columnas de la p-ésima matriz frontera. Sea $n_p = b_{p-1} + z_p$, donde $b_p = \dim(B_p)$ y $z_p = \dim(Z_p)$ (los grupos de imagen y el kernel del homomorfismo ∂_p), de tal forma que a la izquierda tenemos b_{p-1} columnas con un 1 y a la derecha tenemos z_p columnas con ceros. Una vez que tenemos todas las matrices frontera en la forma normal, podemos extraer los números de Betti como diferencias entre las dimensiones, $\beta_p = \dim(Z_p) - \dim(B_p)$ para $p \geq 0$.

Reducción

Para hacer la reducción de ∂_p, procedemos de manera similar a la eliminación Gaussiana para sistemas de ecuaciones lineales. En a lo mas dos operaciones, movemos un 1 de la equina superior izquierda, y con a lo más $n_{p-1} - 1$ sumas de filas y $n_p - 1$ sumas de columnas, hacemos cero el resto de la primera columna y la primera fila. Aplicamos este procedimiento de manera recursiva a la submatriz obtenida removiendo la primera fila y la primera columna. Empezamos la reducción inicializando la matriz a $N_p[i.j] = a_i^j$ para todo i y j, y llamando la función para $x = 1$, la posición del elemento de la diagonal considerado. La estructura del algoritmo se muestra en la Figura 4.

```
void REDUCE(x)
  if there exist k ≥ x, l ≥ x with N_p[k,l] = 1 then
    exchange rows x and k;  exchange columns x and l;
    for i = x + 1 to n_{p-1} do
      if N_p[i,x] = 1 then add row x to row i endif
    endfor;
    for j = x + 1 to n_p do
      if N_p[x,j] = 1 then add column x to column j endif
    endfor;
    REDUCE(x + 1)
  endif.
```

Figure 4: Estructura del algoritmo

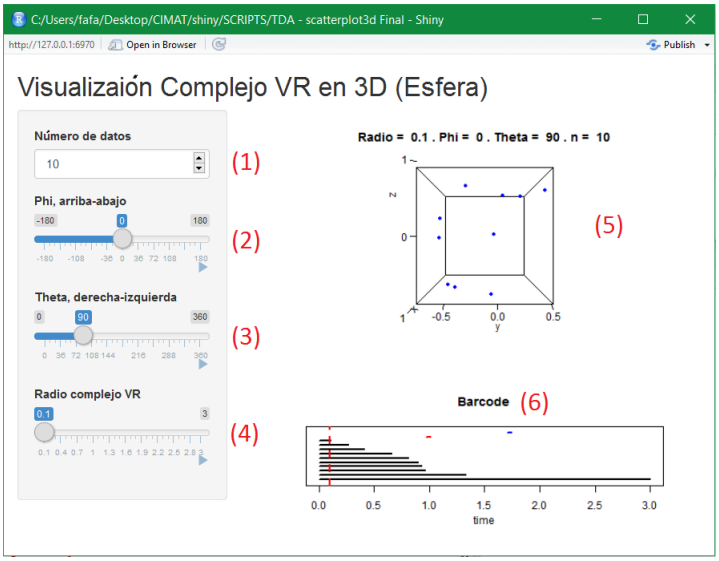
Tenemos a los más n_{p-1} operaciones de filas y n_p operaciones de columnas por llamada recursiva y a lo mas $(n_{p-1} + n_p) \min\{n_{p-1}, n_p\}$ operaciones de filas y columnas en total.

Shiny

Ademas de reproducir los algoritmos para generar los números de Betti, se creó una aplicación utilizando el paquete Shiny en Rstudio.

Shiny es un paquete de R que permite de manera sencilla crear aplicaciones web interactivas (apps) desde RStudio. Toda la documentación y tutoriales para utilizar este paquete están disponibles en shiny.rstudio.com

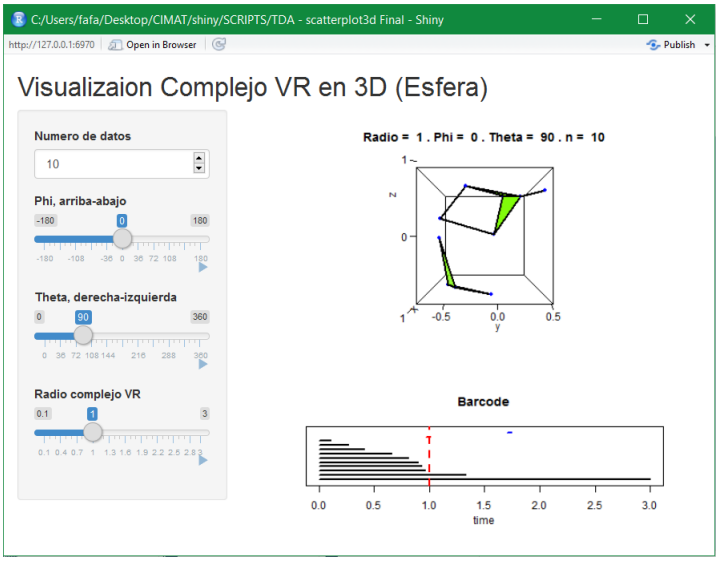
El producto final se muestra a continuación.



El programa tiene las siguientes características:

- Se genera una muestra aleatoria uniforme de tamaño n de una esfera en tres dimensiones. La cantidad n se puede modificar en el campo marcado con (1) en la figura, correspondiente al número de datos.
- La gráfica de los puntos se muestra en la parte superior derecha, marcada con (5).
- La Figura (5) puede rotarse hacia arriba o abajo, modificando el slider (deslizador) marcado con (2), o se puede presionar el icono de "play" en la esquina inferior derecha del slider para obtener una rotación continua.
- La Figura (5) puede rotarse hacia la derecha o izquierda modificando el slider (deslizador) marcado con (3), o se puede presionar el icono de "play" en la esquina inferior derecha del slider para obtener una rotación continua.
- El código de barras asociado a la nube de puntos mostrada en (5) puede verse en la zona inferior derecha marcada con (6). La línea roja indica el tamaño del radio del complejo VR utilizado para graficar la nube de datos.
- El radio del complejo VR puede modificarse con el slider marcado con (4), también se puede presionar el icono de "play" en la esquina inferior derecha del slider para obtener un aumento continuo.

Un ejemplo del mismo complejo mostrado en la figura anterior pero con una radio mayor, el radio aumentó a 1. Se observa tanto en el código de barras como en la gráfica que hay un 1-ciclo.



Ejemplo de uso

A continuación se muestra una manera en que esta aplicación puede ser usada para ilustrar algunos conceptos relacionados con el análisis topológico de datos. Se detallan pasos a seguir con comentarios que pueden ser usados en el salón de clase o durante una exposición introductoria del tema. Los pasos son los siguientes:

1. Inicie la aplicación.
2. Cambie el número de datos a 15 en la casilla marcada con (1).
   (a) Observe la nube de puntos. Haga rotaciones de derecha a izquierda y arriba a abajo utilizando los slider o el icono de "play" en (2) y (3). Comente sobre como estos puntos provienen de una esfera de radio 1.
   (b) Observe el código de barras. Identifique las componentes conexas (en negro), 1-cliclos (en rojo) y 2-ciclos (en azul), de no haber, pruebe aumentando el número de datos a 20. Identifique en el código de barras los radios o tiempos del complejo VR donde se alcanzan los ciclos encontrados.
3. Aumente progresivamente el radio del complejo. Utilizando el icono de "play" en (4). Detenga cuando el complejo alcance un radio de 1.
   (a) Observe el comportamiento de las componentes conexas. Comente sobre la "muerte" de las clases de equivalencia en el grupo de homología H_0.
   (b) Comente sobre la formación de 2-simplejos al conectarse tres aristas que forman un triángulo. Esta es una característica del complejo de VR, es diferente para el complejo de Cech, donde pueden haber tres aristas conectadas formando un triángulo que no esté "relleno" por un 2-simplejo.
4. Cambie el radio del complejo al valor donde ocurre un 1-ciclo.
   (a) Identifique el 1-ciclo en la gráfica, debe verse como un aro o anillo conectado por varios puntos de la nube de datos.
   (b) Obtenga perspectivas diferentes del complejo simplicial seleccionando los ángulos Phi=0,90,180,270.
5. Coloque el radio de VR en 0.1. Aumente ahora el número de puntos en $n = 15, 20, 25, 30, 35, 40$.
   (a) Observe el código de barras en cada etapa. Comente sobre el ruido topólogico y la persistencia. A médida que aumenta el número de puntos en la nube de datos se asemeja a una esfera, los 1-ciclos (en rojo) son cada vez más pequeños, este es el ruido, y los 2-cíclos, presentan una barra que es cada vez más grande, esta es la barra que persiste.
   (b) Comente sobre los números de Betti de una esfera, los cuales son $\beta_0 = 1$, $\beta_1 = 0$, $\beta_2 = 1$, y como el código de barras y la noción de persistencia muestran una aproximación de estos.

Conclusiones

- Se reprodujeron los algoritmos para generar números de Betti. Con este proyecto además de entender la estructura algebraica asociada a los grupos de Homología y su sencillez al trabajarlos como espacios vectoriales se hizo presente la notoria dificultad de trabajar en campos escalares distintos a $\mathbb{Z}_2 = \{0, 1\}$. Además de la interpretación, los algoritmos de reducciones matriciales se dificultan mucho más.
- Una problemática con la aplicación, que ocurre en general con los programas de este tipo, es que aumentar mucho en número de datos o el radio del complejo, implica un costo computacional muy alto. Por lo que se recomienda usarla para pocos datos, no más de 30.
- Se creó una aplicación para visualizar de manera interactiva los complejos de VR. Esta aplicación tiene la gran ventaja de no necesitar conocimiento previo del lenguaje R. Por ello puede ser utilizada por cualquier persona para observar el crecimiento de la filtración en tres dimensiones.
- La idea es que pueda ser mostrado en presentaciones al público para ilustrar la ideas como: nube de datos, código de barras, complejo de VR, componentes conexas, 1-ciclos, 2-ciclos, persistencia, números de Betti, nacimiento y muerte.