Morfismos, Vol. 23, No. 1, 2019, pp. 27–36

Análisis Topológico de Datos y Homología Persistente, usando el lenguaje R *

José de Jesús Aguirre Tepole [1] Angel Emilio de León Gutiérrez [2]

Resumen

Este reporte presenta un caso práctico para el Análisis Topológico de Datos y Homología Persistente, utilizando el lenguaje y entorno de programación R, en especial para la clasificación de imágenes tridimensionales de modelos, aún cuando están en diferentes posturas. Se usa como base el trabajo realizado por Peter Bubenik y Leo Betthauser para la Tercera Escuela de Análisis Topológico de Datos y Topología Estocástica [3].

2010 Mathematics Subject Classification: 55N99, 68W05, 55U99
Keywords and phrases: Análisis Topológico de Datos, ATD, TDA, Homología Persistente, R

1. Introducción

Tomando como base las notas para el taller dadas en [2], se realizaron varios experimentos, modificando las figuras a comparar, cambiando las funciones de filtrado y se dan algunas conclusiones y posibles líneas de experimentación futuras.

Para un breve repaso a las bases teóricas que sustentan estos experimentos, se puede acudir a las siguientes fuentes: [1], [7], [12], [5], entre otras.

* **R es un lenguaje y ambiente para cálculo estadístico y graficación.**

[1] tepole@esfm.ipn.mx.
[2] angel.deleon@banxico.org.mx.

2. Experimentos realizados

Se tienen figuras tridimensionales que han sido escaneadas y convertidas a una representación en caras triangulares y sus correspondientes vértices, si se vuelven a pintar se ven imágenes como la que se muestra en la Figura 1.

Figura 1: Ejemplo de figura escaneada

Se filtran dichos triángulos y se calcula su homología persistente usando el software Perseus [8]. Una vez hecho esto se puede calcular los paisajes de persistencia usando el Persistence Landscapes Toolbox [4]. Finalmente, se comparan las figuras a pares y se realiza una clasificación de las mismas.

En el conjunto de datos con el que se contaba, había imágenes de gatos, perros, caballos, lobos, centauros, gorilas, dos hombres y una mujer. Al momento de realizar pruebas con distintas combinaciones, notamos que sólo en el caso de las dos figuras masculinas tenía problemas para clasificar correctamente, por lo que procedimos a modificar las funciones de filtrado, para ver si cambiaba el escenario.

La función de filtrado original estaba basada en la siguiente:

```
# filtrar por el k-vecino mas cercano
vertex_values <- knn.dist(vertices,k=10)[,10]
```

Esta función básicamente realiza un filtrado de la triangulación que va a generar complejos simpliciales, utilizando la distancia del k-vecino más cercano, usando los 10 vecinos más próximos. Con este filtrado como base, se calculan los valores de los triángulos como:

```
for (i in 1:num_triangles) {
    # asignar a cada triangulo el
    # valor promedio de sus vertices
    triangle_values[i] <-
      mean(vertex_values[triangles[i,]])
}
```

Si visualizamos el orden en el que se van creando los complejos simpliciales, veremos algo similar a lo mostrado en la Figura 2. Si en vez de filtrar por el k-vecino más cercano, se calcula el valor de cada triángulo como:

```
s <- vector(length = num_triangles)
for (i in 1:num_triangles) {
    s[i] <-
      (triangles[i,1] +
        triangles[i,2] +
          triangles[i,3])/2
    triangle_values[i] <- sqrt(
        s[i]*(
          s[i]-triangles[i,1]) +
          s[i]*(s[i]-triangles[i,2]) +
          s[i]*(s[i]-triangles[i,3]))
}
```

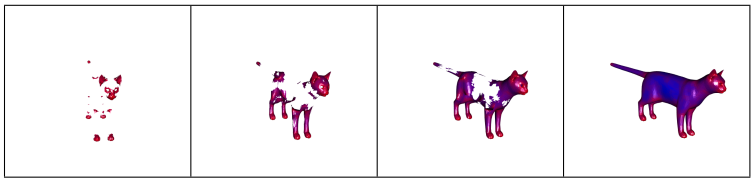

Figura 2: Generación de Complejos Simpliciales

es decir, como el área del triángulo en sí, usando la fórmula de Herón que sólo necesita los lados del triángulo a, b y c y se calcula como

$$\text{Área} = \sqrt{s\,(s-a)\,(s-b)\,(s-c)},$$

donde $s = \frac{a+b+c}{2}$. En este caso, la creación de los complejos simpliciales se da como se muestra en la Figura 3.

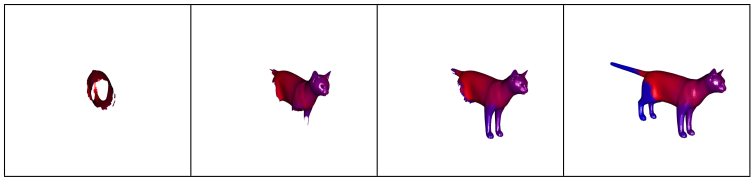

Figura 3: Generación de Complejos Simpliciales, usando la función de área para filtrado

Ahora por último consideremos como función de filtrado el máximo de la coordenada Z de los vértices que conforman cada triángulo, esto se logra con la siguiente función:

```
for (i in 1:num_triangles) {
```

```
    # asignar el maximo de la componente
    # 'z' de los vertices de cada triangulo
    triangle_values[i] <-
      max(vertices[triangles[i,],3])
}
```

Veremos la generación de los correspondientes complejos simpliciales como se ve en la Figura 4

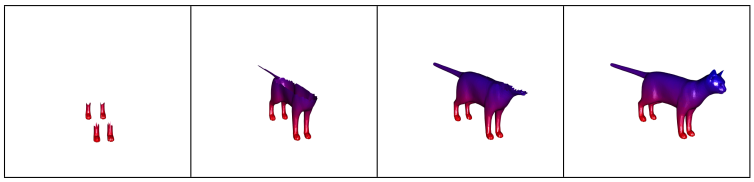

Figura 4: Generación de Complejos Simpliciales, usando max de Z para filtrado

3. Resultados de la clasificación

Aunque el objetivo final del proceso es la clasificación automática de las figuras, podemos notar que hay una enorme flexibilidad para generar los complejos simpliciales y se puede lograr con un mínimo de cambios en el entorno del lenguaje R.

Para cada selección de la función de filtrado, se puede hacer la clasificación usando uno de los siguientes criterios:

1. Clasificación usando SVM[3], kernels [6] y validación k-cruzada [9].
2. Clasificación usando SVM y validación k-cruzada.
3. Clasificación usando PCA[4], SVM y validación k-cruzada.

Para cada criterio, dependiendo de la función de filtrado, da una clasificación con mayor o menor precisión en la predicción de la clase a la que pertenece la figura analizada. Los resultados puntuales están mostrados en las tablas al final, un resumen de los mismos se puede ver en la Tabla 1.

A pesar de que la precisión en la predicción es menor en el caso del área, podemos notar en las Tablas 5, 6 y 7 que las figuras semejantes se confunden más, por ejemplo los perros y gatos y por supuesto las figuras humanas. Esto nos indica que esta función generaliza estas clases y podríamos considerar las clases humano y mascota-doméstica para una mejor clasificación.

[3] Support Vector Machines, o máquinas de vectores de soporte, véase [11]

[4] Principal component analysis, o Análisis de componentes principales, véase [10]

	Criterio 1	Criterio 2	Criterio 3
Media	0.78	0.86	0.83
Área	0.6	0.69	0.66
Máximo	0.54	0.62	0.48

Tabla 1: Precisión en la predicción, usando diferentes funciones de filtrado

4. Conclusiones

Vemos que la forma en la que se realiza el filtrado puede influir grandemente en la fase del aprendizaje automático, sobre todo si se quiere usar el TDA para clasificar conjuntos de datos en las subclases más características.

Para el caso de figuras tridimensionales como las mostradas anteriormente, este análisis puede ser sumamente útil al momento de clasificar los diferentes sujetos de prueba, se pudo observar que con diferentes funciones el porcentaje de aciertos aumentaba y se podía incluso agrupar figuras que aunque diferentes, pertenecían al mismo grupo, por ejemplo todos los humanos.

5. Trabajo Futuro

Se puede extender el análisis para incluir más funciones de filtrado, por ejemplo usar tanto el área como la orientación de cada triángulo para generar los complejos simpliciales, o tomar en cuenta la distancia de cada triángulo al centro de gravedad de la figura.

También se pueden incluir diferentes "expresiones" para cada figura, por ejemplo, que tengan el hocico abierto o cerrado, o mostrando los dientes. Estas variaciones servirían para mejorar la fase de aprendizaje.

Por otro lado, los vértices de los triángulos son puntos en $\mathbb{R}^3$, pero si en vez de ello tuviéramos por ejemplo datos financieros como el monto de un pago, la institución financiera y la fecha, y tuviéramos siempre parejas de puntos que nos indicaran el originador y el beneficiario de un pago, podríamos analizar el flujo del dinero en los sistemas de pagos del país, obteniendo la estructura interna de los mismos, tanto en la dimensión de las instituciones, como en la de los montos, como en las fechas en donde se mueve el dinero. Esto podría generar información importante de la economía del país y nos podría dar idea de los puntos fuertes y débiles de la misma.

6. Apéndice: Tablas de resultados

	cat	david	dog	gorilla	michael	victoria
cat	9	0	0	0	0	0
david	0	3	0	0	6	0
dog	0	0	11	0	0	0
gorilla	0	0	0	21	0	0
michael	0	12	0	0	12	2
victoria	0	0	0	0	2	22

Tabla 2: Filtrado con la media, Primer criterio, Precisión de predicción = 0.78

	cat	david	dog	gorilla	michael	victoria
cat	9	0	0	0	0	0
david	0	8	0	0	4	0
dog	0	0	11	0	0	0
gorilla	0	0	0	21	0	0
michael	0	7	0	0	13	0
victoria	0	0	0	0	3	24

Tabla 3: Filtrado con la media, Segundo criterio, Precisión de predicción = 0.86

	cat	david	dog	gorilla	michael	victoria
cat	9	0	0	0	0	0
david	0	7	0	0	8	0
dog	0	0	11	0	0	0
gorilla	0	0	0	21	0	0
michael	0	8	0	0	11	0
victoria	0	0	0	0	1	24

Tabla 4: Filtrado con la media, Tercer criterio, Precisión de predicción = 0.83

	cat	david	dog	gorilla	michael	victoria
cat	0	0	0	0	0	0
david	0	8	0	0	8	7
dog	6	0	10	0	0	0
gorilla	0	0	0	21	0	0
michael	3	4	0	0	8	4
victoria	0	3	1	0	4	13

Tabla 5: Filtrado con el área, Primer criterio, Precisión de predicción = 0.6

	cat	david	dog	gorilla	michael	victoria
cat	6	0	3	0	0	1
david	1	5	0	0	7	2
dog	2	0	8	0	0	0
gorilla	0	0	0	21	0	0
michael	0	7	0	0	10	2
victoria	0	3	0	0	3	19

Tabla 6: Filtrado con el área, Segundo criterio, Precisión de predicción = 0.69

	cat	david	dog	gorilla	michael	victoria
cat	5	0	2	0	1	2
david	0	3	0	0	5	4
dog	3	0	9	0	0	0
gorilla	0	0	0	21	0	0
michael	0	7	0	0	12	2
victoria	1	5	0	0	2	16

Tabla 7: Filtrado con el área, Tercer criterio, Precisión de predicción = 0.66

	cat	david	dog	gorilla	michael	victoria
cat	5	1	0	1	4	2
david	1	7	1	0	6	5
dog	2	0	8	0	0	0
gorilla	0	0	0	20	0	0
michael	0	1	2	0	2	5
victoria	1	6	0	0	8	12

Tabla 8: Filtrado con el máximo de Z, Primer criterio, Precisión de predicción = 0.54

	cat	david	dog	gorilla	michael	victoria
cat	6	1	1	0	2	1
david	0	5	1	0	7	2
dog	2	0	8	0	1	0
gorilla	0	0	0	21	0	0
michael	1	6	1	0	5	4
victoria	0	3	0	0	5	17

Tabla 9: Filtrado con el máximo de Z, Segundo criterio, Precisión de predicción = 0.62

	cat	david	dog	gorilla	michael	victoria
cat	4	3	2	1	3	0
david	1	3	2	0	9	3
dog	4	0	6	0	1	0
gorilla	0	0	0	20	0	0
michael	0	6	1	0	2	8
victoria	0	3	0	0	5	13

Tabla 10: Filtrado con el máximo de Z, Tercer criterio, Precisión de predicción = 0.48

Agradecimientos

Agradecemos al Dr. Peter Bubenik y a su estudiante de postgrado, M.S. Leo Betthauser por los conceptos y técnicas presentadas en el curso. También a los organizadores por el total apoyo durante el evento.

José de Jesús Aguirre Tepole
Departamento de Matemáticas,
Escuela Superior de Física y Matemáticas,
Instituto Politécnico Nacional,
Av. IPN Edificio 9,
U. P. Adolfo López Mateos,
Ciudad de México, México,
tepole@esfm.ipn.mx

Angel Emilio de León Gutiérrez
Oficina de Desarrollo de Sistemas de Provisión de Liquidez,
Dir. Gral. de Sistemas de Pagos y Servicios Corporativos,
Banco de México,
5 de Mayo 6 - Condesa
Ciudad de México, México,
angel.deleon@banxico.org.mx

Referencias

[1] Peter Bubenik. Introduction to topological data analysis and persistent homology. URL: `http://people.clas.ufl.edu/peterbubenik/files/intro_tda_worksheet.pdf`.

[2] Peter Bubenik. Topological data analysis with r workshop. URL: `http://people.clas.ufl.edu/peterbubenik/files/tda_r_workshop.pdf`.

[3] CONACYT. Tercera escuela de análisis topológico de datos y topología estocástica. URL: `http://atd.cimat.mx/es/tercera-escuela-atd`.

[4] Paweł Dłotko. The persistence landscape toolbox. URL: `https://www.math.upenn.edu/~dlotko/persistenceLandscape.html`.

[5] Herbert Edelsbrunner and John Harer. Persistent homology — a survey. URL: `https://users.cs.duke.edu/~edels/Papers/2008-B-02-PersistentHomology.pdf`.

[6] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. URL: `http://dx.doi.org/10.1214/009053607000000677`.

[7] Michael Lesnick. Studying the shape of data using topology. URL: https://www.ias.edu/ideas/2013/lesnick-topological-data-analysis.

[8] Vidit Nanda. Perseus. URL: http://people.maths.ox.ac.uk/nanda/perseus/index.html.

[9] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. URL: http://leitang.net/papers/ency-cross-validation.pdf.

[10] Lindsay Smith and Jonathon Shlens. A tutorial on principal components analysis. URL: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

[11] Alex Smola and Bernhard Schölkopf. Kernel machines.org. URL: http://www.kernel-machines.org/.

[12] Matthew Wright. Introduction to persistent homology. URL: https://www.youtube.com/watch?v=h0bnG1Wavag.