Topología para Análisis de Datos: Un Caso de Estudio

Diego Gutiérrez Vargas

ABSTRACT

El documento pretende enseñar el uso de técnicas de análisis de datos basados en topología para poder extraer insights diferentes a lo usual con técnicas que dejan de lado la métrica. Teniendo en cuenta, se usa el Mapper para poder hacer un objeto topológico para poder hacer una aglomeración con base a DBSCAN y KMeans.

Keywords: TDA,Topología, Machine Learning

1 PREGUNTAS DE INVESTIGACIÓN Y LITERATURA RELEVANTE

La problemática a analizar es referente al cambio climático, siendo que es un problema que afecta a la población mundial y sigue siendo uno de los mayores desafíos de los humanos en la actualidad. En la base de datos de Kaggle (https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data) por ciudad, se encuentra información sobre su posición geográfica(Longitud y Latitud), la temperatura promedio en el día y la incertidumbre de dicha temperatura, nombre de ciudad y país correspondiente. Siendo que en esta se encuentran ciudades y países de alrededor del mundo, se delimita la investigación a nivel de ciudad.

Las preguntas de investigación consisten en:

1. ¿Se puede apropiadamente separar, de manera topologica las series de tiempo para poder ver comportamientos similares de series de tiempo?

Las técnicas a trabajar de TDA sería las de Mapper, para tener insights de como se podrían agrupar las ciudades en diferentes países. Un caso de uso para está técnica se encuentra con Chen and Volić (2021), quienes usan TDA con el algoritmo de Mapper y DBSCAN para poder hacer un mejor visualizado y modelaje de la epidemia causada por el COVID-19 en los Estados Unidos, en el cuál se logra identificar como se desarrollo la pandemia en el país y regiones donde hubiera prevalencia de infecciones. El Mapper junto con DBSCAN son utilizados por su versatilidad y como se buscan

Otro método para realizar clusterizaciones especificamente de Series de tiempo son exploradas en Aghabozorgi et al. (2015). En esta investigación se pretende expandir en el análisis de series de tiempo en ella se utilizan 4 componentes (Representación de series de tiempo, medidas de similaridad, prototipos de clustering y clustering de series de tiempo)principales para categorizar una serie de tiempo completa.

2 OBJETIVO E HIPÓTESIS.

2.1 Objetivo

El objetivo es poder clasificar diferentes series de tiempo usando las propiedades topologicas usando DBSCAN para encontrar así las ciudades que tienen comportamientos similares en cuánto a su temperatura a lo largo del tiempo acotado.

2.2 Hipotesis inicial

Se puede hacer una separación en base a un indicador de la serie de tiempo para la aglomeración adecuadad de ellas, sin importar la región geográfica en la que están.

3 METODOLOGÍA

3.1 Preprocesamiento

Para poder manejar series de tiempo de las ciudades, se hace lo siguiente:

1. Se eliminan las ciudades que tengan datos faltantes
2. Se acota las series de tiempo de 1993/01/01 - 2013/01/01 (2013 es el último año que tiene información disponible)
3. Se vuelven a eliminar ciudades, en este caso, por tener mismo nombre pero estar en diferentes paises.
4. Se crea un dataframe para poder hacer la alimentación al Mapper, donde se toma la diferencia entre las medias de los primeros 10 años y los últimos 10 años entre la media de los primeros 10 años.

Terminas entonces teniendo 122 ciudades únicas a las cuales no les falta ningún dato en algún tiempo, un indicador que representen las series de tiempo

3.2 Análisis Estadístico

Se encuentran las métricas para cada serie de tiempo en la ciudad: desviación estandár, varianza, mediana, mínimo y máximo y se hace un histograma para cada métrica de las ciudades para ver si hay una distribución prominente para poder hacer alguna separación. Sin embargo, se encuentra en los diferentes histogramas que no hay alguna distribución clara que pueda segregar grupos de series de tiempo, viendosé la necesidad de ahora usar TDA.

3.3 Algoritmos

3.3.1 Algoritmo de TDA: Mapper

Antes de usarlo se usa dos tipos de proyecciones para poder usar la técnica: Isomap y UMAP. El Isomap es una técnica que con base a las distancias geodesicas computa grafos de vecindad, lo que permite mantener las propiedades locales de los datos. Por otra parte, el UMAP se enfoca en mantener la estructura global de los datos

Como el mapper permite hacer una representación gráfica de los datos a una menor dimensión conservando tanto sus propiedades locales como globales por el Isomap y UMAP.

Parametros	Mapper	Parametros	Isomap, UMAP
N. Cubes	[10,15*]	componentes	[3*,40],[2*,3]
Overlap	[0.1,0.25,0.3*]	random state	1
distancia	cosine		

Table 1. Parametrización de Mapper, Isomap y UMAP

3.3.2 Algoritmos de ML: KMeans y DBSCAN

La técnica de KMeans es muchas veces la técnica a implementar cuando se hace algún tipo de clusterización por ofrecer resultados rápidos y eficientes para hacer aglomeraciones de datos. Sin embargo, se sabe que es sensible al ruido y como sus puntos iniciales son generados de manera aleatoria también tiene un factor estocástico que afecta en el desempeño de la clusterización de datos. Es usado más que nada para ofrecer una comparativa al DBSCAN y ver la técnica que agrupa de mejor manera las distintas ciudades seleccionadas. Además, se usa la métrica de Silhouette y técnica de codo para obtener su K óptimo, teniendo así un método base a mejorar.

El DBSCAN es un algoritmo de clusterización de Machine Learning no supervisado muy utilizado para todo tipo de aplicaciones y el cuál a diferencia de K-Means no contiene un factor estocástico. Por naturaleza no es sensible al ruido y hace usualmente un buen trabajo con datasets que lo contienen.

Parametros	DBSCAN	Parametros	KMeans
Min. samples	[5,10*]	K	[4*,13]
Eps	[0.5,1*]	Max. Iter.	300
Distancia	Coseno	Random State	0

Table 2. Parametrización de Técnicas de Clusterización

4 RESULTADOS

4.1 Parametrización de Algoritmos

Se realizó diferentes iteraciones para encontrar un DBSCAN que pudiera maximizar los nodos conectados. En KMeans, se realizó el ”Silhouette Score”, el cuál obtuvo un 0.8 y ”Elbow Method” para tener un K óptimo, sin embargo, esto diferían claramente del número, por lo cuál se realizo la clusterización con ambos números obtenidos. Se terminó descartando la clusterización con k=13, dado a que se obtenían muchos nodos solitorias que aislaban a muchas ciudades sin proporcionar algún insight de como pudieran estar relacionadas con otras.

Para la parametrización de DBSCAN, se vario al igual los diferentes componentes en Isomap, UMAP, eps y samples mínimos, el cuál se termino quedando con un mínimo de 10 y una distancia de 1 en eps para hacer mayores conexiones entre nodos, siendo que los anteriores proporcionaban visualizaciones se resultaban únicamente viendo nodos con una sola conexión como máximo y algunos sin conexión.

4.2 Clusterización de ML

La versión final de DBSCAN que guarda mayor información se tiene que se forman 5 componentes conexas a las cuales solo dos tienen un único nodo, aislando su comportamiento de cualquier otro tipo de serie de tiempo.

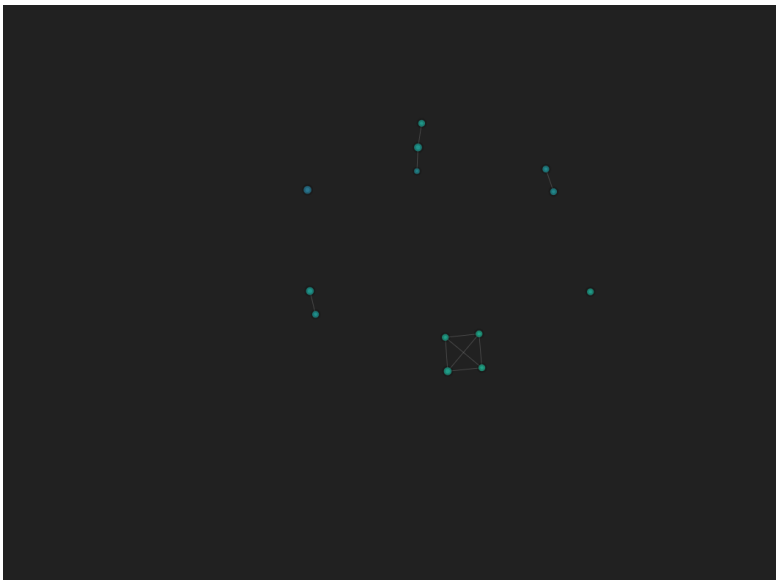

Figure 1. Clusterización basado en DBSCAN.

Por otra parte, KMeans logra tener más segmentaciones de los nodos, sin embargo, esto no significa una mejor clusterización de los puntos, sino simplemente una mayor segmentación de los datos.

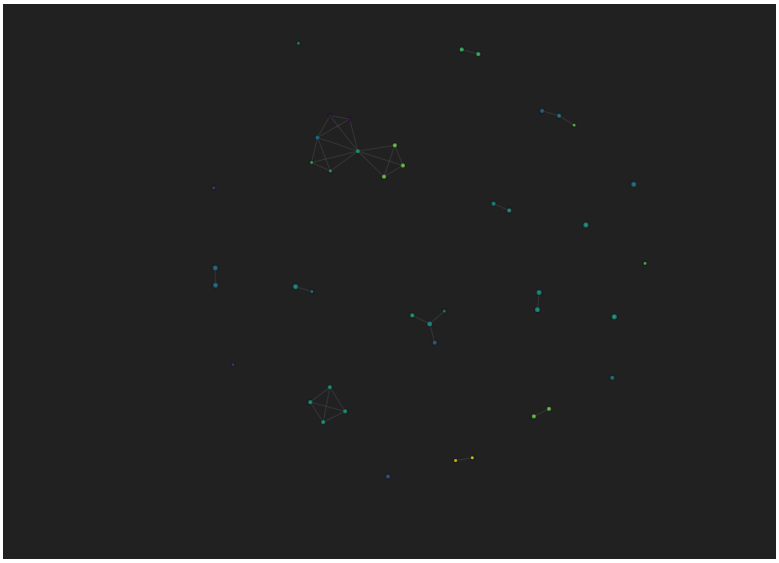

Figure 2. Clusterización basado en KMeans.

4.3 Análisis de Intra cluster

Investigando más a detalle en los clusters se encuentran varios clusters que incluyen solo una ciudad. Para el KMeans se fue viendo que había muchos grupos que se repetían entre los nodos y estos eran precisamente los nodos con mayores conexiones. Además, como KMEans no tiene paramétro mínimo de datos se encontraban nodos con 3 o 4 miembros. Con una simple visualización de los nodos(cubo1 y cubo2), se puede ver que si tiene menores miembros pero los agrupa de manera adecuada con base a la proyección e indicador dado.

En el caso de DBSCAN, hacía aglomeraciones de ciudades con mayores miembros con 20 o más miembros por nodos y por grupo de nodos, los cuales preservan en su mayoría esos comportamientos de picos, e incluso algunos con misma magnitud y comportamiento. Visualizando nodos conectados(cubo11 y cubo12, se puede ver que los comportamientos de la serie de tiempo son suficientemente cercanas en comportamiento como para juntarlas en una.

REFERENCES

Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering–a decade review. *Information systems*, 53:16–38.

Chen, Y. and Volić, I. (2021). Topological data analysis model for the spread of the coronavirus. *PLOS ONE*, 16(8).