

# Una metodología de *clustering* para agrupar series temporales en regiones contiguas

Pablo Quintana\*

## Resumen

Los algoritmos de *clustering* con restricción espacial existentes están enfocados principalmente en obtener regiones contiguas, pero no hay muchos trabajos que contemplen la agrupación de series temporales. En este *paper* se propone una nueva metodología de regionalización que contempla los avances que se vienen haciendo en el campo de redes neuronales y aprendizaje no supervisado denominada *spatial constraints into deep embedding clustering* (SDEC). El propósito del algoritmo es agrupar unidades espaciales transformando el espacio de atributos en otro de menor dimensión representado por variables latentes que logren destacar la característica del grupo al que pertenecen. Se realizaron simulaciones de series temporales por cada unidad espacial en dónde, SDEC muestra resultados superiores con respecto a otros métodos existentes en la literatura. Con el objetivo de aplicar el método a problemas de la vida real, se hace un breve estudio de la evolución del virus del Covid 19 por departamento o partido de la República Argentina, posteriormente se trata de llegar a una interpretación de los resultados obtenidos para evaluar el funcionamiento del algoritmo.

**Clasificación JEL:** C18, C21, R12.

**Palabras Clave:** Clustering espacial, Clustering espacio-temporal, Regionalización

---

\*Facultad de Ciencias Económicas, UNCUYO, Doctorado en Economía; Centro Universitario (M5502JMA), Ciudad de Mendoza, Argentina; email: pabanib@hotmail.com

# 1. Introducción

Las metodologías de *clustering* se encuentran dentro de las herramientas de aprendizaje automático, más precisamente del aprendizaje no supervisado. Este tipo de aprendizaje pretende reconocer patrones entre variables para poder agrupar las unidades de análisis en una cantidad desconocida de grupos presentes en la población. Es decir, estas metodologías no cuentan con información sobre la cantidad de grupos, como si sucede bajo el aprendizaje supervisado. La importancia de estas técnicas se debe a que se considera que este campo representa la mejor aproximación sobre cómo aprendemos los seres humanos (LeCun, Bengio e Hinton 2015). Como inconveniente, este tipo de problemas es considerado *NP-Hard* (coloquialmente, un sistema cuya complejidad algorítmica es intratable y su solución no es necesariamente óptima, véase Cook 2006), por lo cual no solo dificulta encontrar una solución, sino que también se vuelve complejo saber si la solución hallada es óptima.

En economía regional, el análisis de *clustering* se ha trabajado en la literatura con el objetivo de construir regiones que sean contiguas y muestren un alto grado de similitud en el conjunto de atributos. Los nombres que adquiere este tipo de problemas en la literatura son diversos, entre ellos podemos mencionar a *clustering* condicional, *clustering* con relaciones restringidas, *clustering* restringido, *clustering* contiguo y restringido, *clustering* regional, regionalización o *clustering* restringido por conectividad (ver Duque, Church y Middleton 2011). La resolución de este problema consiste en agregar unidades espaciales a una región en donde las unidades que pertenezcan a ella sean lo más homogéneas posibles y exista un alto grado de heterogeneidad con las unidades de las otras regiones. Todo esto considerando la contigüidad espacial que va estar dada principalmente por la matriz de contigüidad espacial  $C$  de dimensión  $n \times n$  donde  $n$  son las unidades espaciales, que relaciona cada unidad espacial con todas las unidades que son vecinas tal que  $c_{ij} = 1$  si  $i$  y  $j$  son vecinas, y  $c_{ij} = 0$  si  $i$  y  $j$  no lo son. La regionalización es una técnica de interés en varias disciplinas, entre ellas se incluyen la distribución de escuelas, redistribución de políticas y control policial por áreas. Pero, además, los problemas socio económicos generalmente son procesos fuertemente influenciados por la composición del espacio urbano lo que posiciona a la regionalización como una herramienta cada vez más útil para detectar esos patrones.

Por otro lado, las técnicas de clustering también son usadas en agrupar series de tiempo. El objetivo en estos casos es tratar de juntar aquellas unidades que tengan la realización de sus respectivas variables en el tiempo de manera similar. Lo importante en el desarrollo de métodos de clustering pasa por la definición de la medida de distancia o de disimilitud. Por lo tanto, en clustering temporal suele usarse las mismas herramientas modificando la función objetivo para que considere los efectos temporales. Una forma de verlo es considerando la distancia euclídea solo que en problemas temporales se considera cada periodo de tiempo como si fuera una variable, una medida de disimilitud más efectiva, aunque más lenta es la denominada «*dynamic time warping*» DTW (Sakoe y Chiba (1978) y Faloutsos, Ranganathan y Manolopoulos (1994)). Por otro lado, en Huang y col. (2016) utilizan la distancia euclídea ponderada de acuerdo a la estampa de tiempo y de esta manera suavizar la línea temporal.

El objetivo de este trabajo es ensamblar técnicas de aprendizaje no supervisado para resolver el problema de encontrar regiones en las cuales todas las unidades tengan similitud en el desarrollo de sus series temporales. Considerando que la evolución en el tiempo de una variable no es independiente periodo a periodo, se trata de encontrar variables ocultas en los datos que nos permitan generar los mismos datos observados, está técnica de reducción de la dimensión es conocida en la literatura como *autoencoders*. Reduciendo la dimensión y considerando la agrupación por variables latentes se pueden obtener resultados superiores al agrupamiento simple de los datos.

Este algoritmo fue presentado por Xie, Girshick y Farhadi (2016) al cual denominaron DEC. Pero este tipo de agrupación no funciona en el armado de regiones que tengan contigüidad en el espacio, el cuál es el objetivo de los métodos con restricción espacial, por lo tanto, en el presente trabajo se propone una serie de ajustes a la metodología DEC para poder aplicarlo a problemas espacio-temporales.

## 2. Revisión de la literatura

Los algoritmos de *clustering* tienen como objetivo reducir el número de observaciones. Básicamente tratan de agrupar  $n$  observaciones en  $p$  clústeres. Los grupos que así se forman tienen que mantener la mayor semejanza posible entre los elementos que se hayan en el interior y diferenciarse de los otros grupos. Estos objetivos se transmiten a los algoritmos a través de una minimización o maximización, depende el caso, de lo que se denomina función de disimilitud. Similarmente, la regionalización busca encontrar grupos de datos similares pero con contigüidad en el espacio, o sea encontrar vecindarios que muestren similares características entre sí. Este tipo de trabajos se estudia desde hace tiempo, pero por problemas computacionales es en los últimos años en que ha tomado más auge.

Existen alternativas simples para combinar métodos de *clustering* tradicionales y técnicas espaciales. Una de ellas es la implementada por (Anselin, Syabri y Kho 2010) mediante el programa GeoDa en donde se agregan las coordenadas geográficas como dos atributos adicionales a las unidades espaciales, tal que la función objetivo considera esta información como cualquier otro atributo. Otra alternativa similar es la propuesta por Webster y Burrough 1972, Wise, Haining y Ma 1997, Haining, Wise y Ma 2000 y más recientemente Yuan y col. 2015; Cheruvilil y col. 2017; Chavent y col. 2018, en donde permiten diferenciar la información espacial mediante una ponderación diferente. En este caso, la información espacial está representada por una matriz de contigüidad que contiene pesos espaciales, representada habitualmente con la letra  $W$ , marcando una diferencia respecto a los atributos regulares que se están estudiando. Sin embargo, en ambas estrategias, la inclusión de la información espacial no asegura que los grupos se encuentren espacialmente contiguos.

Para considerar contigüidad, entonces, hay que adoptar la restricción que produce la matriz de contigüidad de manera estricta, o sea, agrupar solo aquellas unidades que pertenezcan a un mismo vecindario. Openshaw 1977 plantea el problema de la zonificación automatizada (*automatic zoning problem*), en este caso la metodología de agrupamiento (AZP) si consideraba la restricción espacial de manera estricta. El problema parte de asignar una cantidad  $n$  de áreas en  $p$  regiones contiguas. La solución consiste en un procedimiento iterativo que parte de una partición inicial  $P_p^0$  para luego intercambiar las áreas entre las regiones vecinas considerando una función objetivo global que indicará si los cambios producen mejoras. Esta metodología demanda muchos recursos computacionales y puede caer fácilmente en soluciones locales que no permitan que el algoritmo llegue a una solución óptima. La solución de este problema llega después en Openshaw y Rao 1995 dónde plantean mecanismos de búsqueda local como son *tabu search* y *simulated annealing*.

La metodología anterior obtiene muy buenos resultados, pero tiene el problema de la elección de la cantidad de regiones tiene que ser elegida de manera arbitraria por el investigador. Es por eso que en Duque, Church y Middleton 2011 y Duque, Anselin y Rey (2012) se incrementa la complejidad del problema agregando a la cantidad de grupos de manera endógena al algoritmo. Esta metodología, denominada Max-p, lo que hace es encontrar cuál es la cantidad óptima de grupos que permita cumplir con un umbral establecido en base a una variable poblacional (por ejemplo, cantidad de personas por región). Una vez que encuentra una partición con esta característica, se

procede a mejorar la agrupación de manera similar a que lo hace AZP. La particularidad de que se determine la cantidad de regiones de manera endógena lo hace un algoritmo interesante que ha tenido adaptaciones como se puede ver en Li, Church y Goodchild (2014), She, Duque y Ye (2017) y Ye, She y Benya (2018).

Siguiendo las metodologías jerárquicas, también se logran adaptar los métodos para llegar a solucionar el problema de la restricción espacial. La metodología denominada *spatial constraints into an agglomerative hierarchical clustering* (SCHC) que se puede ver en Murtagh (1985) y Gordon (1996) entre otros, utiliza la construcción de árboles jerárquicos restringidos a un grafo que lo proporciona la matriz de contigüidad. De esta forma obtenemos una regionalización que cumple con los requisitos de contigüidad. Más adelante, Assunção y col. (2006) proponen un procedimiento espacial denominado *SKATER (Spatial Kluster Analysis by Tree Edge Removal)*. El método consiste en utilizar el árbol de expansión mínimo (*minimum spanning tree*, MST en siglas) para luego cortar las relaciones entre unidades espaciales que mejoren la función objetivo. El MST consiste en conectar todas las unidades en un grafo que pasa por cada nodo una sola vez, por lo tanto, el grafo contiene  $n$  nodos y  $n - 1$  ejes. Es por eso que cada vez que se corta un eje, el árbol se divide en dos. Este proceso durará hasta encontrar el número de regiones buscadas. Otro método, siguiendo la línea de los métodos jerárquicos es el método denominado REDCAP (*regionalization with dynamically constrained agglomerative clustering and partitioning*) propuesto por Guo 2008. Esta metodología consiste en complementar los métodos jerárquicos con un tratamiento diferencial de la contigüidad espacial. En realidad, es una combinación de seis metodologías, cuatro formas de enlazar las unidades y dos formas de tratar la contigüidad espacial. La ventaja que tiene el trabajar con estas metodologías es que no requiere tantos recursos computacionales como si lo hacen AZP y Max-p.

Por otro lado, el agrupamiento de series temporales ha tenido desafíos diferentes y más que nada pasan por la forma de medir la distancia entre dos secuencias. Cuando trabajamos en el *clustering* de series temporales, lo que habitualmente se busca es que coincidan las curvas en el tiempo, o sea que dentro del mismo grupo tengamos un mismo comportamiento en el tiempo. Las mediciones más populares son la distancia euclídea se puede ver en Faloutsos, Ranganathan y Manolopoulos (1994) o la *dynamic time warping (DTW)* Sakoe y Chiba (1978). Esta última en realidad no es una distancia, pero es muy útil para medir la disimilitud de dos series temporales, aunque es un proceso más complicado desde el procesamiento computacional. Adaptaciones de los algoritmos de clustering se han utilizado en la literatura para tomar en cuenta las agrupaciones temporales, generalmente tratan de transformar el espacio y capturar las formas de las series temporales, esto se puede ver en Huang y col. (2016) dónde tratan de suavizar las secuencias y agrupar con la lógica de *k-means*. En Paparrizos y Gravano (2015) el objetivo es destacar las formas de las series temporales para luego agruparlas.

### 3. Metodología utilizada

El objetivo del algoritmo que se presenta en esta sección tiene como objetivo determinar regiones en las que variables han seguido las mismas tendencias mostrando un comportamiento similar y a la vez, diferente de otras regiones. El hecho de establecer regiones con la restricción de contigüidad favorece a la visualización y puede servir para la aplicación de medidas determinadas, que ya se harán a través de regiones definidas por sus características y no solo por definiciones políticas como es habitual en los estudios regionales.

### 3.1. Problema espacio - temporal

El problema que se intenta resolver surge cuando tenemos áreas o unidades espaciales, que son las unidades básicas de análisis, con una serie de variables que representan secuencias temporales. En el marco de este artículo denominaremos atributos a esas variables, regiones a un conjunto de áreas con características similares y pertenencia al mismo vecindario y territorio al conjunto de todas las áreas.

Sea  $A = \{A_1, A_2, \dots, A_n\}$  el conjunto de todas las áreas con  $n = |A|$ .

Los atributos del área  $i$ -ésima se representan como  $A_{iy}$ , en donde  $y \in Y = \{1, 2, \dots, m\}$ , siendo que cada área tiene un vector de atributos  $\mathbf{X}_i = \{a_{i1}, \dots, a_{im}\}$  en donde  $a_{iy} = (y_1, y_2, \dots, y_T)$  es una secuencia temporal de posibles valores del atributo  $A_{iy}$ .

Para establecer relaciones entre áreas definimos una función de disimilitud  $d : A \times A \rightarrow \mathbb{R} \cup \{0\}$  que se basa en el conjunto de atributos  $Y$ . Dicha función  $d_{ij} = d(A_i, A_j)$  debe satisfacer que  $d_{ij} \geq 0, d_{ij} = d_{ji}$ , y  $d_{ii} = 0, \forall i, j = 1, 2, \dots, n$ . Alternativamente, pueden definirse funciones de distancias.

Sea  $G = (V, E)$  el grafo o red asociado con  $A$  en donde el vértice  $v_i \in V$  corresponde al área  $A_i \in A$  y el eje  $\{v_i, v_j\} \in E$  si y solo si las áreas  $A_i$  y  $A_j$  comparten un borde en común. Nótese que el grafo  $G$ , en este caso, es similar a la matriz  $C$ , definida en la sección previa, que relaciona con un 1 si  $i$  y  $j$  son áreas vecinas y con un 0 en caso que no lo sean.

Sea  $P_p = \{R_1, R_2, \dots, R_p\}$  una partición de las áreas  $A$  dentro de  $p$  regiones con  $1 \leq p \leq n$ . Denominamos a  $\Pi$  como el conjunto de todas las posibles particiones de  $A$ .

La partición óptima  $P_p^*$  será aquella que consiga la mayor similaridad posible de los atributos  $\mathbf{X}_i$  y  $\mathbf{X}_j$  para dos áreas cualesquiera  $A_i, A_j \in R_j$ . La similaridad en este contexto pasa por buscar una distancia entre matrices, ya que el espacio de características que tiene cada área es una matriz  $t \times m$ . La forma más sencilla de realizar dichas comparaciones, es con la transformación de las matrices en un vector de dimensión  $t \times m$  y aplicar distancia euclídea entre ellos. Otra alternativa, es utilizar el DTW, el uso de esta medida no permite trabajar los datos como si fuera un vector, esto tiene un costo computacional más alto que aplicado de la otra forma.

Otra manera de realizar las comparaciones de similitudes es aplicando transformaciones al espacio de características y encontrar un espacio de variables latentes. Para estos casos, cada área transformará sus atributos en un vector de variables latentes  $\mathbf{h}_i$  que va tener una dimensión menor a la de los vectores originales. La distancia ahora, se calcula sobre estos vectores  $d_{ij} = d(\mathbf{h}_i, \mathbf{h}_j)$  esto se explica en la siguiente sección de manera más clara. La transformación del espacio de atributos es la técnica que se usa en el presente *paper*, aunque la manera de computar las distancias se realiza bajo otras características que serán explicadas en la sección que sigue.

Encontrar la partición  $P_p^*$  óptima es un problema de características NP-hard y por ellos hay que recurrir a simplificaciones del problema que nos permitan acercarnos a una solución, pero en consecuencia con este tipo de problemáticas, ni siquiera se puede concluir con claridad si la solución obtenida está cerca del ideal. Por eso es que ninguna de las técnicas existentes se puede definir como mejor que otra.

### 3.2. Variables latentes

Cuando observamos los datos, los atributos de las áreas los hemos definidos como secuencias temporales. Por lo tanto, la realización de un atributo en un periodo  $t$  no es independiente de la realización de ese mismo atributo en el periodo  $t - 1$ , este hecho nos permite suponer que todos los datos que tenemos existen en un espacio de dimensión inferior. Dicho de otra manera, existen  $h$  variables latentes con  $h < Y * T$  que pueden explicar el comportamiento de cada área. Esto habilita

al uso de técnicas de reducción de la dimensión, las mismas son metodologías de aprendizaje no supervisado que tienen como objeto encontrar un espacio subyacente de menor dimensión que proporciona la mayor cantidad de información necesaria para entender el comportamiento de los datos. Si bien la técnica más conocida para la reducción de la dimensión es el análisis de componentes principales (ACP), actualmente existen herramientas más avanzadas que permiten mayor flexibilidad y pueden establecer relaciones no lineales, entre ellas podemos mencionar t-SNE (Maaten e Hinton 2008) y los *autoencoders* (AE). Para una buena base de estos últimos es recomendable leerlo desde Goodfellow, Bengio y Courville (2016). En base al funcionamiento de estos dos algoritmos se plantea un mecanismo de *clustering* espacialmente restringido.

## Autoencoders

Los AE son herramientas de aprendizaje no supervisado que han empezado a tener auge con el crecimiento del *deep learning* y las redes neuronales. A diferencia de una red neuronal supervisada que trata de predecir una variable objetivo, en los AE tanto el *input* como el *output* es el mismo espacio. Esto se logra con la construcción de una red neuronal de dos etapas, primero una que reduzca la dimensión del espacio de entrada y se produzca lo que habitualmente se denomina cuello de botella. A esta etapa la denominaremos *encoders* (codificador). Luego a partir de este resultado, se establece una capa que trata de llegar al espacio original con el cual se había comenzado. Esta segunda etapa se denomina *decoder* (decodificador). Para verlo más formalmente podemos seguir el *paper* de Vincent y col. 2010:

*Encoders*: busca mapear una función  $f_{\theta}$  que transforma un vector de entrada  $\mathbf{x}$  dentro de una representación  $\mathbf{y}$

$$f_{\theta}(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (1)$$

en dónde el parámetro es  $\theta = \{\mathbf{W}, \mathbf{b}\}$ , dónde  $\mathbf{W}$  es una matriz de pesos de tamaño  $d' \times d$ ;  $\mathbf{b}$  es un vector de dimensionalidad  $d'$  y  $s$  es una función no lineal, generalmente denominadas función de activación en la jerga de las redes neuronales.

*Decoders*: El resultado que arroja la etapa de encoders  $\mathbf{y}$ , denominado capa oculta, variables ocultas o variables latentes. Es luego mapeado para reconstruir un vector  $\mathbf{z}$  d-dimensional en el espacio de entrada,  $\mathbf{z} = g_{\theta'}(\mathbf{y})$ .

$$g_{\theta'}(\mathbf{y}) = s'(\mathbf{W}'\mathbf{y} + \mathbf{b}'), \quad (2)$$

con apropiado tamaño de parámetros  $\theta' = \{\mathbf{W}'; \mathbf{b}'\}$

En general  $\mathbf{z}$  no debe ser interpretado como una reconstrucción exacta de  $\mathbf{x}$ , pero en términos de probabilidad cómo de sus parámetros la distribución  $p(X|Z = \mathbf{z})$  puede generar a  $\mathbf{x}$  con una alta probabilidad. Nótese que, si se eliminan las funciones no lineales, obtenemos las funciones que se utilizan en ACP, por lo tanto, lo que diferencia a los AE es precisamente la introducción de la no linealidad.

Las funciones con las que se trabaja, generalmente son de difícil interpretación, pueden tener muchas capas que estén conectadas entre sí, pero esto nos permite tener mucha flexibilidad a la hora de trabajar con los datos que se quieren agrupar. Por eso podemos trabajar con datos temporales incluso de varias variables y con datos espaciales, la herramienta nos permite ir diferenciando los tratamientos todo dentro del mismo algoritmo.

### 3.3. *Deep embedding clustering* espacialmente restringido

La metodología denominada *deep embedding clustering (DEC)* es un método de *clustering* que se basa en el uso de AE, fue presentado por Xie, Girshick y Farhadi (2016). Los resultados obtenidos por la misma superaron a la de otras metodologías de agrupamiento existentes. El algoritmo en si no está destinado a series temporales y menos a construir regiones que sean contiguas. Una utilización de este método en un problema espacio-temporal es la que aplican Asadi y Regan (2019) en la que agrupan datos del tránsito. A continuación, presentamos una extensión de la metodología DEC para poder ser aplicada a problemas con restricción espacial y que tengan una correlación temporal.

El método consiste en tres etapas. La primera etapa es la de trabajar con el espacio de variables latentes, para esto es necesario practicar un entrenamiento de los *autoencoders* que nos permita encontrar las variables latentes más representativas de los datos. Como vemos, ahora la representación temporal de los atributos de cada área se verá reducida a una representación de estas variables ocultas, esto es  $h_i = (a_{i1}, a_{i2}, \dots, a_{ih})$  en donde cada  $a_{iy}$  representa al valor de la función  $f_\theta(x_i)$  para el *encoder*  $y$ . Esto ya se muestra diferente a las otras metodologías ya que ahora estamos ante un espacio transformado y es el que se va agrupar. La técnica de reducir el espacio a variables latentes para luego agrupar es utilizada en la literatura, tanto con ACP, t-SNE y AE, en estos casos, luego de obtener a  $h$  se procede a agrupar de acuerdo alguna otra metodología conocida.

La segunda etapa consiste en inicializar el algoritmo agrupando de acuerdo método *k-means* en base al espacio de características latentes. Esta etapa nos arroja una partición inicial  $P_p = \{R_1, R_2, \dots, R_p\}$  en la que a cada región se le puede representar con su centroide  $\{\mu_1, \mu_2, \dots, \mu_p\}$ . Minimizar la distancia de cada una de las áreas hasta el respectivo centroide de la región a la que pertenece es equivalente a decir que la región es homogénea.

Por cada unidad espacial con la que estamos trabajando podemos asignarle una probabilidad de pertenecer a cada una de las regiones basada en la distancia que existe entre el espacio de variables latentes del área y el centroide de las regiones  $d(h_i; \mu_j)$ . De esta forma, obtenemos una asignación débil de las unidades espaciales a las regiones. La probabilidad que habitualmente se usa es la que surge del método t-SNE y es una distribución t-student como se muestra a continuación.

$$q_{j|i} = \frac{(1 + \|h_i - \mu_j\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{h \neq i} (1 + \|h_i - \mu_h\|^2/\alpha)^{-\frac{\alpha+1}{2}}} \quad (3)$$

siendo  $\alpha$  los grados de libertad, que habitualmente se usa un solo grado de libertad.

Como en toda metodología de aprendizaje automático necesitamos obtener una función de pérdida que sea derivable para que le permita al algoritmo adaptar los pesos e ir mejorando los resultados. En este caso utilizaremos la divergencia de Kullback-Liebler que se representa de la siguiente forma:

$$L = KLIC(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} = \sum_{i \neq j} [p_{ij} \log p_{ij} - p_{ij} \log q_{ij}] \quad (4)$$

Falta determinar la probabilidad objetivo  $P$ , para poder aplicar la función de pérdida correctamente. La elección de esta probabilidad es crucial para el buen rendimiento del algoritmo y para obtener efectos de contigüidad en el espacio. Por lo tanto, es necesario que se busque una probabilidad que fortalezca la predicción, ponga mayor énfasis en aquellos puntos que han sido asignados con una mayor confiabilidad y por último que normalice la contribución a la función de pérdida para evitar que grandes grupos distorsionen el espacio de variables ocultas. Siguiendo a

Xie, Girshick y Farhadi (2016) se consideró una función de probabilidad que eleva la probabilidad  $q_i$  al cuadrado y luego se normaliza por la frecuencia por clúster:

$$\bar{p}_{ij} = \frac{q_{ij}^2/g_j}{\sum_k q_{ij}^2/g_k} \quad (5)$$

siendo  $g_j = \sum_i q_{ij}$  la frecuencia suave del clúster.

Para considerar el efecto espacial, se realizó una modificación a la probabilidad objetivo establecida previamente. Cómo se busca que las regiones sean contiguas, entonces la probabilidad de que una unidad pertenezca a esa región también va depender de la probabilidad de que las regiones vecinas pertenezcan a la misma región. Es por esto que para obtener la probabilidad objetivo  $P$  consideraremos la probabilidad calculada de los vecinos que se establezcan de acuerdo a la matriz de contigüidad, esto para cada una de las unidades espaciales. Entonces a la probabilidad establecida anteriormente se le agrega la probabilidad de que tienen todas las demás unidades de pertenecer al mismo grupo multiplicada por el grafo espacial determinado.

$$p_{ij} = \bar{p}_{ij} + \frac{\left(\sum_k q_{kj}^2 C_{ik}\right)/g_j}{\sum_k q_{ij}^2/g_k}, \quad (6)$$

en dónde  $q_{kj}^2$  es la probabilidad de la unidad  $k$  de pertenecer al grupo  $j$ ,  $C_{ik}$  representa al elemento  $ik$  de la matriz de contigüidad que se esté trabajando para el caso. Nótese que, si la matriz utilizada es una de contigüidad, esto hará que solo se sumará la probabilidad de los vecinos de la unidad. Un resultado interesante es que en las pruebas, el algoritmo mostró mejores resultados solo considerando la probabilidad de los vecinos y excluyendo  $\bar{p}_{ij}$ .

La tercera etapa consiste en la optimización de la función de pérdida  $L$  adaptando progresivamente las probabilidades recientemente planteadas. La forma de optimización se realiza con las mismas técnicas utilizadas para optimizar redes neuronales habituales, la más utilizada es el descenso del gradiente, pero existen adaptaciones del mismo, la elección del método de optimización es un hiperparámetro que se puede ir ajustando mientras produzca mejoras en los resultados.

El DEC espacialmente restringido (SDEC) no deja de ser una red neuronal con la última capa adaptada para la regionalización. Es por eso que la forma en que se van actualizando los parámetros internos se hace siguiendo el mecanismo de *back propagation*, esto implica que la red se va ajustando desde el final hacia atrás. Los centroides  $\{\mu_j\}$  correspondientes a cada región son los primero en adaptarse, ya que las unidades que componen a cada región son diferentes cada vez, la regla de actualización va ser la que sigue el camino contrario al gradiente de la función de pérdida. Luego se procede a la optimización de los pesos y sesgos necesarios para la determinación de los *encoders*  $s(\mathbf{W}\mathbf{x} + \mathbf{b})$ . La cantidad de capas que puede contener el *encoders* es otro hiper parámetro y se debe ir adaptando al problema que se está trabajando.

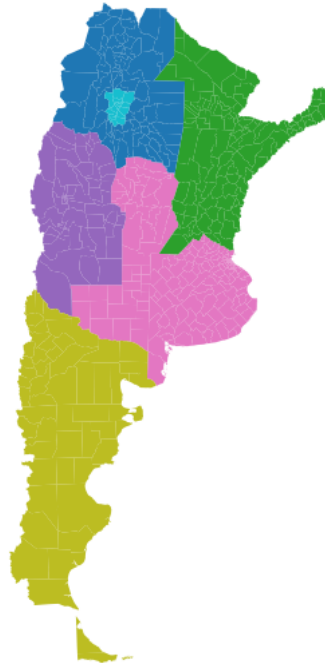
La probabilidad  $Q$  es la que realiza una asignación débil de las áreas a las regiones, pero en este trabajo se busca una metodología con asignación fuerte a cada región, por lo tanto, hay que buscar una regla para hacerlo. La estrategia de los autores del algoritmo DEC planteado es simplemente definir los grupos de acuerdo a la probabilidad  $q$  más grande. Para garantizar aún más la contigüidad espacial, la propuesta en este trabajo es utilizar una de las metodologías de clustering espacial ya utilizadas en la literatura, solo que, en vez de agrupar las características originales, se encargarán de agrupar de acuerdo a las variables ocultas que arrojan los encoders. El utilizado en el presente paper es el método jerárquico con la restricción espacial que se viene trabajando. Esto no solo arroja el resultado final, sino que también sirve como medida para parar el algoritmo antes en caso de que se llegue a situaciones en que no mejora la asignación.



## 4. Simulaciones

En esta sección se realizó una comparación entre las metodologías de *clustering* con restricción espacial que existen en la literatura y la metodología que se propone en este trabajo. La contextura espacial se sacó conectando los departamentos y partidos de la Argentina, y se establecieron seis regiones con comportamientos distintos entre sí y similares en la región. El problema que se está considerando es la evolución de un indicador en el tiempo, por lo tanto, para las simulaciones se consideró dos variables que van variando a lo largo de cien periodos cada una. Se consideró una función cuadrática negativa que depende del periodo de tiempo, por cada una de las seis regiones, distinguiéndolas por su punto máximo, tanto en periodo como en valor, y por los cortes al eje de abscisas (momento en que empieza el fenómeno y que termina). Las unidades espaciales adoptan la función de la región a la que pertenecen. El objetivo perseguido en la simulación es el que se puede ver en la imagen 1.

Figura 1: Mapa objetivo



Para agregar ruido al sistema por cada unidad se agrega un componente aleatorio que proviene de una distribución uniforme centrada en cero y tiene una amplitud de 0.5 (adopta valores de -0.25 a 0.25) en el primer caso y de uno en el segundo (adopta valores de -0.5 a 0.5). En este ejemplo tratamos de simular proporciones (valores que van de 0 a 1), por lo que se agrega una restricción más que es que los valores no pueden ser menores a cero. Las metodologías utilizadas para hacer las comparaciones son: método jerárquico con restricción espacial, calculada desde la librería *sklearn* de *python*; el método SKATER, REDCAP y SCHC utilizando la librería *rgeoda* para R. Todos los métodos arrancaron con el objetivo de seis regiones y con la parametrización estándar para cada uno. Para la metodología de clustering SDEC se desarrolló en Python con la librería Keras y *sklearn* principalmente, el código del trabajo se puede ver en: [https://github.com/pabanib/clustering\\_espacial](https://github.com/pabanib/clustering_espacial). Se parte de seis autoencoders y seis regiones y se consideró como método espacial el jerárquico con restricción espacial ya utilizado. En todos los casos, se utilizó una matriz de  $k$  vecinos más cercanos considerando seis vecinos.

## Características SDEC

La conformación de una red neuronal para construir el algoritmo SDEC deja posibilidades de adaptación de acuerdo al problema planteado. En el caso que se trabaja, tanto en las simulaciones como en el caso empírico de la siguiente sección, se separan los atributos de las áreas en la capa de *input*, ya que solo son dos que tienen varios periodos en el tiempo. Esto significa que la red neuronal adapta los pesos por cada serie temporal por separado en las primeras capas y luego las junta para formar la salida de las variables latentes pretendidas. La cantidad de *encoders* se adapta a la cantidad de grupos que se quiere obtener, esto quiere decir que  $p = h$  y todas las pruebas se han realizado de esta forma.

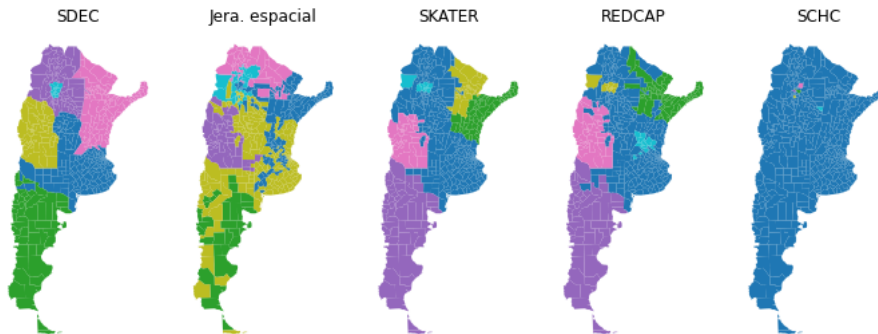
## Resultados simulaciones

Para evaluar las metodologías se utilizaron varias métricas de rendimiento utilizadas en la literatura. En el cuadro 1 se puede ver el resultado que arrojaron cada una de las metodologías en las métricas información mutua ajustada, información mutua, homogeneidad y *rand score*. Como se ve, SDEC se muestra claramente superior a los otros algoritmos presentados, llegando a resultados casi perfectos. En la figura 2 se pueden visualizar los resultados de los cinco métodos utilizados, en este aspecto el mapa construido por el algoritmo SDEC es casi idéntico al objetivo buscado que se visualiza en la figura 1. Estas gráficas junto con los indicadores pueden mostrarnos que, para esta simulación, SDEC está descubriendo por sí solo el comportamiento existente en los datos.

Cuadro 1: Métricas Extrínsecas

	SDEC	Jerarq. esp.	skater	redcap	schc
Adj Mutual Inf	0,9622	0,5711	0,6300	0,6064	0,0264
Mutual Inf	1,5225	0,9011	0,9124	0,8648	0,0377
Homog	0,9597	0,5680	0,5751	0,5451	0,0238
Rand score	0,9890	0,7635	0,7621	0,7530	0,2559

Figura 2: Resultados sobre primera simulación



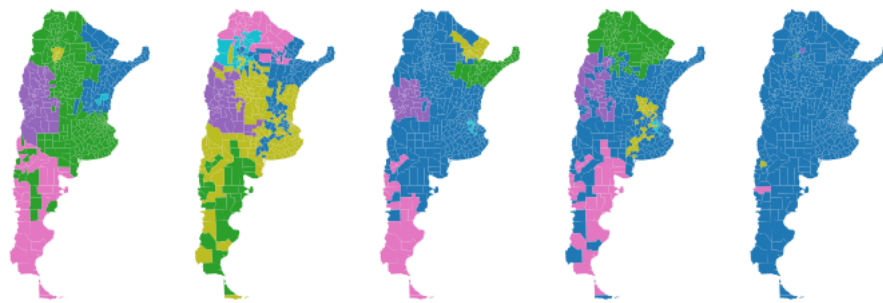
En el segundo caso, con mayor varianza, el rendimiento de las metodologías lógicamente disminuye, pero muestra el mismo patrón anterior. Nuevamente, SDEC se muestra mejor en todas las métricas calculadas (cuadro 2). En la figura 3 se observa la estructura de los mapas que cada uno de los algoritmos elabora. Para este caso el parecido con el mapa original es mucho más confuso que en el caso anterior, pero aun así se puede distinguir un patrón más claro en el algoritmo SDEC. De todas formas, los indicadores de rendimiento se encargan de determinar la eficiencia del método,

en los cuales se logran resultados más que aceptables considerando que el ruido es bastante fuerte para estas simulaciones.

Cuadro 2: Métricas extrínsecas caso 2

	SDEC	Jerarq. esp.	skater	redcap	schc
Adj Mutual Inf	0,7248	0,4481	0,4041	0,3729	0,0169
Mutual Inf	1,0171	0,7020	0,5359	0,5441	0,0289
Homog	0,6411	0,4425	0,3378	0,3430	0,0182
Rand score	0,8086	0,7584	0,5793	0,6660	0,2522

Figura 3: Resultados de la segunda simulación



De acuerdo a los resultados mostrados en estas simulaciones podemos ver que la idea de trabajar con variables latentes que reflejen las características principales de los eventos es satisfactoria. El entrenamiento de los *autoencoders* y la posterior maximización de la probabilidad de pertenencia hace que el algoritmo SDEC elimine todo el ruido aleatorio y por lo tanto no informativo que tienen los datos. De esta forma, la utilización de este tipo de metodología permite detectar algún patrón temporal que este explicando la evolución de los datos.

## 5. Aplicación empírica

La aplicación empírica se realizó con la evolución de los contagios y fallecidos por el virus del Covid en la Argentina. Se tomó como atributos la proporción de ambas variables sobre el total de la población en cada departamento o partido en el periodo que va desde marzo 2020 a julio 2021. Para no tener características que se destaquen por el resto de las otras, al aplicar metodologías de clustering es aconsejable estandarizar los datos. Esta base de datos adopta características similares a las bases de datos simuladas anteriormente, esto nos permite utilizar el mismo armado de la red neuronal de SDEC que se había construido para las simulaciones.

Una de las limitaciones que se presenta al aplicar las metodologías de clustering es que no sabemos la cantidad de grupos utilizar, pero la literatura de clustering tiene algunas técnicas como son el gráfico de codo (*elbow plot*). Esta fue la forma que se adoptó para definir la cantidad de grupos a utilizar para aplicar la metodología.

SDEC consta de tres etapas para su cálculo. Lo primero que se realiza es un entrenamiento de los *autoencoders* para obtener los codificadores. Para este caso se trabaja con una red neuronal que considera, en principio, ambas variables (contagiados y fallecidos) por separado y luego se concatenan previo a determinar los *encoders*. Posterior a eso se trata de decodificar todo en conjunto

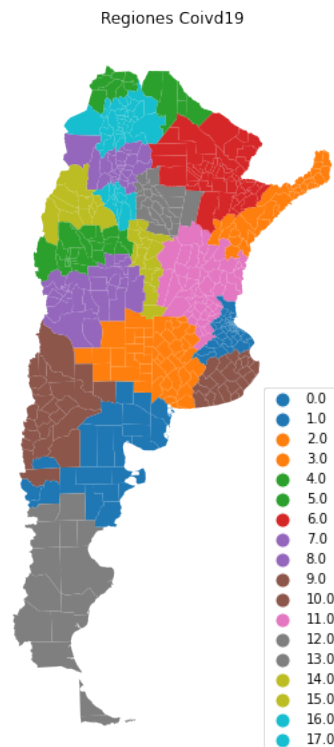
tratando de obtener la matriz de entrada como ya ha sido explicado. Las formas en que se pueden elegir las redes neuronales que constituye los autoencoders son muchas lo que hace difícil determinar cuál sería la composición óptima, pero si se puede seguir criterios del investigador para reducir las posibilidades. En este caso, la elección de tomar las variables por separado al principio, es para que el algoritmo capte primero los efectos temporales de las variables y luego considere el efecto entre ambas.

Luego de finalizado el entrenamiento de los autoencoders, se utilizan las variables codificadas para hacer un agrupamiento inicial. En el caso trabajado se utiliza la metodología k-means para esta inicialización con los parámetros por defecto que ofrece la librería sklearn. Por último, se optimiza esa agrupación inicial con la metodología jerárquica restringida a una matriz de k vecinos más cercanos.

En este caso, en que no sabemos cuál es la verdadera agrupación, no podemos utilizar las métricas extrínsecas, ya que estas se comparan con la correcta agrupación.

Una parte interesante que arroja SDEC es que las regiones logradas, tienen una cantidad de áreas parecidas entre sí, como habíamos visto era una de los objetivos que se buscaba al elegir la función de probabilidad objetivo, por lo tanto, las variables codificadas eliminan el ruido que puede proporcionar un *outlier*. En la imagen 4 se puede ver el resultado que arroja la metodología empleada, se ve que las regiones permanecen contiguas y todas las regiones poseen una proporción pareja de unidades espaciales dentro de ellas.

Figura 4: Resultado SDEC para covid



## 5.1. Una forma de interpretar los resultados

Para entender un poco mejor los resultados de las regiones, pasaremos hacer el análisis con el coeficiente de localización. La particularidad que tiene el coeficiente de localización es que resalta las diferencias existentes entre las regiones al ser una comparación de proporciones. Si una región es homogénea y por lo tanto no existen diferencias significativas en las áreas que las componen, entonces la esperanza de la variable en un área tiene que ser igual a la proporción poblacional  $\frac{t_i}{T}$  multiplicada por la realización de la variable en toda la región  $S_k$ . Esto es:  $E(s_{ik}) = S_k \frac{t_i}{T}$  (ver Billings y Johnson 2012). Por lo tanto, si el área es homogénea a la región nos queda que el coeficiente de localización es igual a uno ya que lo podemos escribir de la siguiente forma:

$$LQ_{jk} = \frac{s_{jk}}{\lambda_{jk}}, \quad (7)$$

dónde  $s_{jk}$  es el valor de la variable  $k$  en el área  $j$  y  $\lambda_{jk} = S_k \frac{t_i}{T}$ .

Si la región constituida es homogénea entonces el coeficiente de localización dentro de ella debe ser igual a uno en todas las áreas y por lo tanto podemos medir la homogeneidad de la siguiente forma:

$$Homog_r = \frac{1}{np} \sum_{i=1}^n \sum_{k=1}^p (s_{ik} - \lambda_{ik})^2, \quad (8)$$

dónde  $p$  es la cantidad de variables que utilizamos, en el caso de estudio sería contagiados y fallecidos en cada uno de los periodos de tiempo.

El índice de homogeneidad lo vamos a obtener sumando la ecuación (8) calculada para cada una de las regiones que obtuvimos con el algoritmo SDEC y lo vamos a dividir por la homogeneidad considerando a todo el territorio en conjunto, para relativizar los valores y tener un índice que vaya de 0 a 1. Aunque pueden existir particiones que hagan este indicador mayor a uno, no sería el caso si los algoritmos funcionan bien.

$$HomogLQ = \frac{1}{R} \sum_{r=1}^R \frac{Homog_r}{HT}, \quad (9)$$

dónde  $R$  es la cantidad de regiones en que se partió el territorio y  $HT$  es la homogeneidad de la ecuación 8 sin considerar la partición realizada.

Los resultados se pueden ver en el cuadro 3, se decidió presentar la homogeneidad de los contagiados y fallecidos por separado, lo que muestra que, para el caso de fallecidos, las regiones construidas con el algoritmo tienen mayor similitud en sus departamentos. El indicador muestra que la partición tiene mejor similitud que considerando a toda la Argentina en conjunto.

Por otro lado, nos queda definir si las regiones son heterogéneas entre sí, que es otro de los objetivos que se busca con la regionalización. Como dijimos el LQ es un buen indicador para resaltar diferencias o concentraciones, por lo tanto, lo seguimos utilizando, pero ahora lo calculamos a nivel región con relación a todo el territorio. No se puede decir que el algoritmo logró regionalizar correctamente con solo calcular el LQ, pero si podemos calcular un intervalo de confianza del mismo por cada región y cada variable considerada y comparar cuantos intervalos tienen intersección distinta de cero como medida de heterogeneidad.

$$LQ_g = \frac{1}{R(R-1)m} \sum_{i=1}^R \sum_{j \neq i}^R \sum_{y=1}^p I(intR_{iy} \cap intR_{jy} \neq \emptyset), \quad (10)$$

en dónde  $R$  es el número de grupos o regiones,  $intR_i$  es el intervalo de confianza de la región  $i$  y  $p$  es la cantidad de variables consideradas, en este caso trabajado también es la cantidad de periodos por cada variable. Este indicador va estar entre 0 y 1 considerando al valor 1 como indicador que no existe ninguna región disjunta de las otras.

En el cuadro 3 se muestran los resultados de este indicador con dos criterios diferentes para constituir el intervalo de confianza. Una forma de hacerlo es considerando una varianza teórica del coeficiente establecida por Moineddin, Beyene y Boyle (2003) y luego se considera la normalidad de la distribución. Esta varianza está fuertemente influida por el número de unidades, por lo tanto, cuando este es grande los intervalos de confianza disminuyen fuertemente y tenemos que la mayoría de las regiones se encuentran separadas entre sí. Es por eso que el índice considerando solo los fallecidos es mayor cuando era más homogéneo que el de contagiados, esto se debe al problema que tiene la varianza. La otra forma es considerando *bootstrap* en cada región de las áreas que lo contienen. En este caso se pasó región por región y periodo por periodo y se intercambió la variable y la población de las áreas por una cantidad de mil veces. Cada vez que se alteraba la composición de áreas se suman todos los valores de las variables y todas las poblaciones y se calculaba el LQ, luego el intervalo fue construido con percentiles al cinco por ciento. En este caso los resultados no se muestran tan bien separados y si muestran que, en el caso de las aglomeraciones de fallecidos, hay mejores diferencias entre regiones.

Cuadro 3: Homogeneidad e índice de localización

	Contagiados	Fallecidos	Total
HomogLQ.	0.3456	0.2688	0.3072
Índice LQ (m)	0.0777	0.2061	0.1419
Índice LQ (b)	0.5279	0.4913	0.5096

Ver los resultados de manera gráfica no es sencillo porque tenemos muchos periodos y muchas regiones, para reflejar los resultados se muestra el LQ por cada periodo que pertenecen a las regiones 0, 3, 9, 11 y 15 las cuales son todas contiguas en el espacio como se ve en la figura 4. En el gráfico 5 se puede observar la evolución del coeficiente de localización para las cinco regiones en cuestión, tanto de la variable contagiados (izquierda) y fallecidos (derecha). La distinción más notoria la tiene la región 0 que arranca con un nivel de contagiados y fallecidos muy distinto al resto y luego se empareja con lo demás. Aunque el nivel de fallecidos continuó alto. Las otras regiones mantienen comportamientos más similares, aunque se pudo ver cambios en las tendencias en los periodos 6 a 11, en las que algunas crecen más rápido en esos periodos como son las regiones 9, 11 y 15 y luego tienen distintos periodos en que se relajan. Por su parte la región 3 tiene un crecimiento más lento pero sostenido en el tiempo. En lo que es fallecidos, los comportamientos son similares, salvo que las distancias entre las curvas son más amplias.

En el gráfico 6 se muestra los intervalos de confianza calculados con *bootstrap* para las mismas cinco regiones de la zona centro del país. Al incluir el intervalo de confianza del LQ podemos observar si en los periodos observados, las regiones son significativamente distintas. En estos casos los intervalos son bastantes amplios lo que muestra pocas regiones que estén aisladas. En los periodos que van del 8 al 16 casi que encontramos a todas las regiones con un coeficiente muy similar entre ellas. Por ejemplo, la región 11 y 9 muestran muchas similitudes en todos los periodos, siendo que sus intervalos de confianza no logran separarse en ningún momento. Pero, por el contrario, podemos ver que la región 0 y 15 se muestran distantes en todos los periodos en lo que respecta a fallecidos, mostrando una verdadera diferencia en sus comportamientos. Además, esta región 15

en los últimos periodos de fallecidos muestra que claramente se ubicó por debajo de todas las otras zonas.

Sin ánimo de profundizar mucho más en el análisis del comportamiento del Covid, el objetivo ha sido tratar de entender como ha funcionado la metodología SDEC. Según estos indicadores intrínsecos el método ha funcionado a medias, tal vez separando más regiones de las que debía, pero si observamos las curvas se muestran comportamientos diferenciados, aunque sus intervalos coincidan. Este tipo de indicador no nos dice si estamos cerca de la solución, solo nos muestra algunos patrones.

Figura 5: Evolución del LQ para la región pampeana

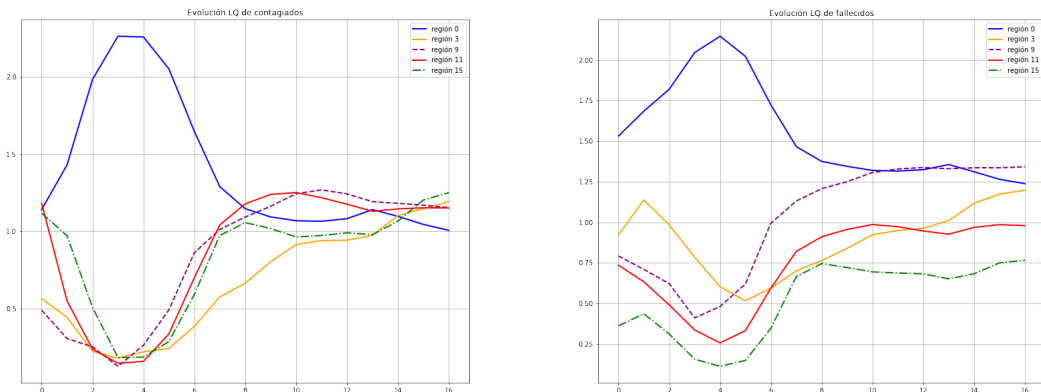
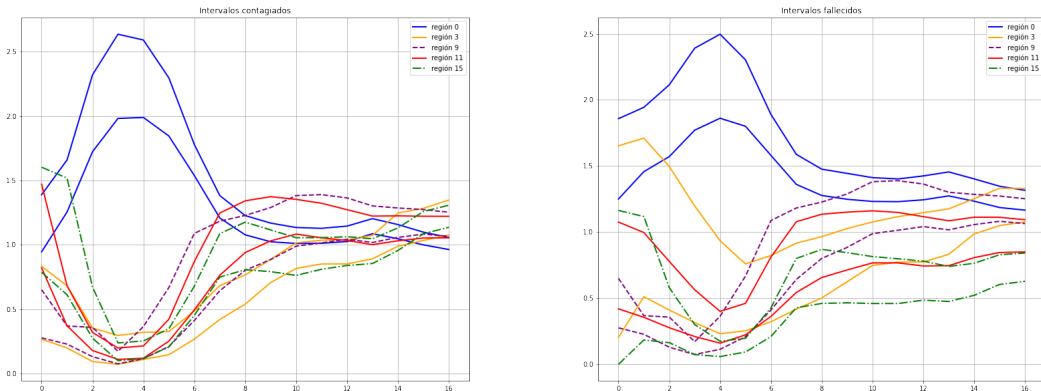


Figura 6: Intervalos de confianza de LQ



## 6. Conclusiones

En el presente *paper* se ha presentado una novedosa metodología de regionalización que es una extensión del método de *clustering* DEC. Esta metodología se basa otras técnicas de aprendizaje no supervisado que están en constante crecimiento últimamente como son los denominados *autoencoders*. El potencial que presentan estas novedosas herramientas se vuelve cada vez más importante en el análisis de realidades más complejas, es por eso que en este artículo se ha trabajado para su aplicación en el campo de la economía regional.

La distribución de los datos en el territorio por lo general no es aleatoria sino que es influida por las distancias espaciales presentes. El método SDEC funcionó mejor que otros métodos existentes en la literatura en los datos simulados que se presentaron, lo que refleja la capacidad de SDEC para detectar patrones de comportamientos en las regiones desconocidas a primera mano. En los casos planteados, se enfatizó en dinámicas temporales de las variables, pero no necesariamente la metodología estaría enfocada en este tipo de problemas, sino más bien en dónde existe alta correlación.

En la aplicación empírica hacer las evaluaciones se hace más complejo, tanto por la cantidad de variables presentadas como para reflejar los resultados de manera gráfica, como en los indicadores que por lo general no reflejan un resultado claro. De todas formas, de acuerdo a lo que se pudo observar, se notaron indicios de que el algoritmo detectó patrones típicos en las regiones que conformó. Cabe aclarar, que el caso del fenómeno covid, se ha comportado de manera homogénea en casi todo el territorio lo que complica el rendimiento de este tipo de metodologías. Pero, en definitiva, la forma de evaluar el rendimiento de la metodología es una tarea pendiente para seguir trabajando.

## Referencias

- Anselin, Luc, Ibnu Syabri y Youngih Kho (2010). "GeoDa: an introduction to spatial data analysis". En: *Handbook of applied spatial analysis*. Springer, págs. 73-89.
- Asadi, Reza y Amelia Regan (2019). "Spatio-temporal clustering of traffic data with deep embedded clustering". En: *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*, págs. 45-52.
- Assunção, R. y col. (2006). "Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees, sapatially". En: *International Journal of Geographical Information Science* 20.7, págs. 797-811.
- Billings, Stephen B y Erik B Johnson (2012). "The location quotient as an estimator of industrial concentration". En: *Regional Science and Urban Economics* 42.4, págs. 642-647.
- Chavent, Marie y col. (2018). "ClustGeo: an R package for hierarchical clustering with spatial constraints, sapatially". En: *Computational Statistics* 33.4, págs. 1799-1822.
- Cheruvilil, Kendra Spence y col. (2017). "Creating multithemed ecological regions for macroscale ecology: Testing a flexible, repeatable, and accessible clustering method, sapatially". En: *Ecology and evolution* 7.9, págs. 3046-3058.
- Cook, Stephen (2006). "The P versus NP problem". En: *The millennium prize problems*, págs. 87-104.
- Duque, Juan, Luc Anselin y Sergio Rey (2012). "The Max-P regions problem". En: *Journal of Regional Science* 52.3, págs. 397-419.
- Duque, Juan, Richard Church y Richard Middleton (2011). "The p-Regions Problem, sapatially". En: *Geographical Analysis* 43.1, págs. 104-126.
- Faloutsos, Christos, Mudumbai Ranganathan y Yannis Manolopoulos (1994). "Fast subsequence matching in time-series databases". En: *ACM Sigmod Record* 23.2, págs. 419-429.
- Goodfellow, Ian, Yoshua Bengio y Aaron Courville (2016). *Deep learning*. MIT press.
- Gordon, AD (1996). "A survey of constrained classification". En: *Computational Statistics & Data Analysis* 21.1, págs. 17-29.
- Guo, Diansheng (2008). "Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP), sapatially". En: *International Journal of Geographical Information Science* 22.7, págs. 801-823.



- Haining, Robert, Stephen Wise y Jingsheng Ma (2000). “Designing and implementing software for spatial statistical analysis in a GIS environment”. En: *Journal of Geographical Systems* 2.3, págs. 257-286.
- Huang, Xiaohui y col. (2016). “Time series k-means: A new k-means type smooth subspace clustering for time series data”. En: *Information Sciences* 367, págs. 1-13.
- LeCun, Yann, Yoshua Bengio y Geoffrey Hinton (2015). “Deep learning”. En: *nature* 521.7553, págs. 436-444.
- Li, Wenwen, Richard Church y Michael Goodchild (2014). “An extendable heuristic framework to solve the p-compact-regions problem for urban economic modeling, sapatially”. En: *Computers, Environment and Urban Systems* 43, págs. 1-13.
- Maaten, Laurens Van der y Geoffrey Hinton (2008). “Visualizing data using t-SNE.” En: *Journal of machine learning research* 9.11.
- Moineddin, Rahim, Joseph Beyene y Eleanor Boyle (2003). “On the location quotient confidence interval”. En: *Geographical analysis* 35.3, págs. 249-256.
- Murtagh, Fionn (1985). “A survey of algorithms for contiguity-constrained clustering and related problems”. En: *The computer journal* 28.1, págs. 82-88.
- Openshaw, S. y L. Rao (1995). “Algorithms for Reengineering 1991 Census Geography”. En: *Environment and Planning A: Economy and Space* 27.3. PMID: 12346252, págs. 425-446. DOI: 10.1068/a270425. eprint: <https://doi.org/10.1068/a270425>. URL: <https://doi.org/10.1068/a270425>.
- Openshaw, Stan (1977). “A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling, sapatially”. En: *Transactions of the institute of british geographers*, págs. 459-472.
- Paparrizos, John y Luis Gravano (2015). “k-shape: Efficient and accurate clustering of time series”. En: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, págs. 1855-1870.
- Sakoe, Hiroaki y Seibi Chiba (1978). “Dynamic programming algorithm optimization for spoken word recognition”. En: *IEEE transactions on acoustics, speech, and signal processing* 26.1, págs. 43-49.
- She, Bing, Juan Duque y Xinyue Ye (2017). “The network-max-P-regions model, sapatially”. En: *International Journal of Geographical Information Science* 31.5, págs. 962-981.
- Vincent, Pascal y col. (2010). “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” En: *Journal of machine learning research* 11.12.
- Webster, R y P Burrough (1972). “Computer-Based soil mapping of small areas from sample data: I. Multivariate Classification and Ordination”. En: *Journal of Soil Science* 23.2, págs. 210-221.
- Wise, Steve, Robert Haining y Jingsheng Ma (1997). “Regionalisation tools for the exploratory spatial analysis of health data, sapatially”. En: *Recent developments in spatial analysis*. Springer, págs. 83-100.
- Xie, Junyuan, Ross Girshick y Ali Farhadi (2016). “Unsupervised deep embedding for clustering analysis”. En: *International conference on machine learning*. PMLR, págs. 478-487.
- Ye, Xinyue, Bing She y Samuel Benya (2018). “Exploring regionalization in the network urban space, sapatially”. En: *Journal of Geovisualization and Spatial Analysis* 2.1, pág. 4.
- Yuan, Shuai y col. (2015). “Constrained spectral clustering for regionalization: Exploring the trade-off between spatial contiguity and landscape homogeneity, sapatially”. En: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, págs. 1-10.