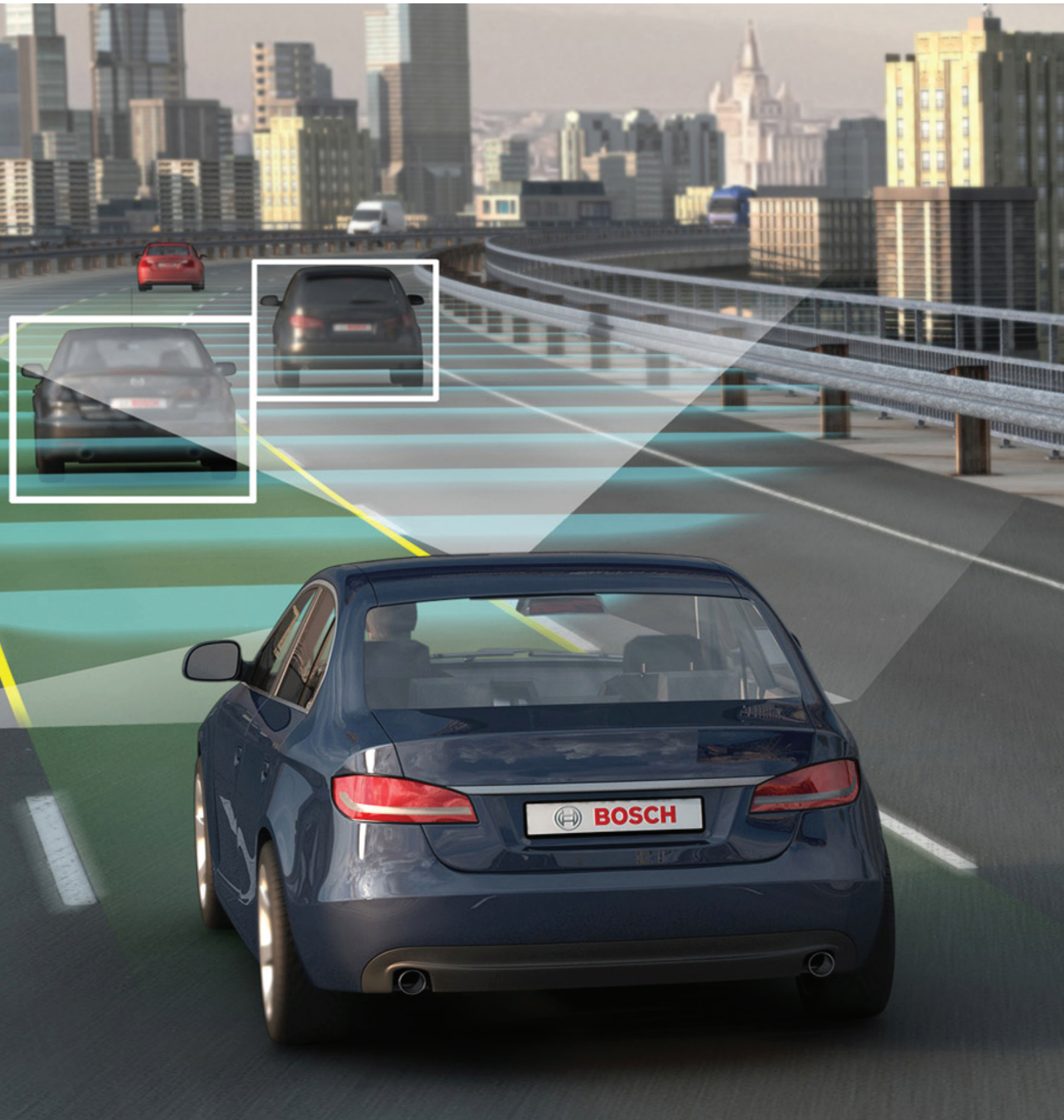
compAS

Identificación de patrones en trayectorias vehiculares aplicando algoritmos de machine learning



Identificación de patrones en trayectorias vehiculares aplicando algoritmos de machine learning

Autores:

CERVANTES SUAREZ CARLOS ANDRÉS
JENIFFER BONILLA BERMEO
CHRISTOPHER CRESPO LEON

Identificación de patrones en trayectorias vehiculares aplicando algoritmos de machine learning

Autores

CERVANTES SUAREZ CARLOS ANDRÉS
JENIFFER BONILLA BERMEO
CHRISTOPHER CRESPO LEON

Primera edición: enero 2018

Diseño de portada y diagramación:
Grupo Compás
Equipo Editorial
ISBN: 978-9942-770-35-6

AGRADECIMIENTO

A Dios, por otorgarme salud, por poner en mi camino a personas maravillosas, de las cuales he aprendido mucho y me han servido como referentes para tomar impulso en seguir aprendiendo y así lograr las metas propuestas.

ÍNDICE GENERAL

ABREVIATURAS

ABP	Aprendizaje Basado en Problemas
UG	Universidad de Guayaquil
FTP	Archivos de Transferencia
g.l.	Grados de Libertad
Html	Lenguaje de Marca de salida de Hyper Texto
http	Protocolo de transferencia de Hyper Texto
Ing.	Ingeniero
CC.MM.FF	Facultad de Ciencias Matemáticas y Físicas
ISP	Proveedor de Servicio de Internet
Mtra.	Maestra
Msc.	Master
URL	Localizador de Fuente Uniforme
www	world wide web (red mundial)
RNA	Redes Neuronales Artificiales
SOM	Self Organizing Maps
BMU	Best Matching Unit
QE	Quantization Error
WCSS	Within Cluster Sum of Squares
KDD	Knowledge Discovery in Databases
GUI	Graphical User Interface
IDE	Entorno de Desarrollo Integrado

SIMBOLOGÍA

s	Desviación estándar
e	Error
E	Espacio muestral
E(*Y*)	Esperanza matemática de la v.a. y
s	Estimador de la desviación estándar
e	Exponencial

ÍNDICE DE CUADROS

ÍNDICE DE GRÁFICOS

PRÓLOGO

Prólogo

Los autores de este libro realizan una que tiene como objetivo comprender el algoritmo de mapas auto – organizados (SOM) a través de la experimentación en diferentes bases de datos científicas para identificar patrones en trayectorias vehiculares GPS. La metodología se basa en el uso de las herramientas que provee la investigación científica, tales como la observación, la experimentación y la hipótesis. Además se aplicó una metodología cascada, ya que se siguió un enfoque secuencial durante el desarrollo de la investigación. Las experimentaciones se realizaron en base al algoritmo de mapas auto-organizados en combinación con k-means y el hierarchical clustering, los cuales fueron implementados en el lenguaje de programación R, con el ID RStudio. El test de hipótesis fue realizado utilizando RCommander, la cual es una herramienta estadística que provee el IDE. Se realizó la validación del algoritmo según las métricas de calidad que posee dicho algoritmo. Posterior a esto se realiza la interpretación de los resultados obtenidos, esto para detectar patrones, inmersos en los datos. Las variables utilizadas para tal efecto fueron la velocidad del vehículo y la hora en cual estaba transitando el mismo. Finalmente se establece las conclusiones, sobre de cuál combinación de algoritmos tuvo un mejor performance según las métricas consideradas, los patrones detectados y se da recomendaciones para investigaciones futuras.

INTRODUCCIÓN

En la actualidad, la información se ha convertido en un recurso muy valioso para la sociedad. La información espacial ha aumentado considerablemente en las organizaciones, haciendo necesaria la explotación de dicha información por medio de la minería de datos. El uso de dispositivos GPS y otros dispositivos de detección de localización para captar la posición de objetos en movimiento es cada vez mayor, y se hace necesario el uso de herramientas para el análisis eficiente de un gran volumen de datos referenciados en el espacio y tiempo (Pedreschi, 2008) (F. Giannotti, 2007) (G. Andrienko, 2007).

La Minería de Datos (Piatesky-Shapiro & Frawley, 1991), se define como el proceso completo de extracción de información, que se encarga además de la preparación de los datos y de la interpretación de los resultados obtenidos, a través de grandes cantidades de datos, posibilitando de esta manera el encuentro de relaciones o patrones entre los datos procesados.

La aplicación de las redes neuronales artificiales (RNA), ha resultado ser una técnica fructífera para la extracción de patrones a partir de conjuntos de datos grandes y complejos. Las RNA son algoritmos informáticos que simulan las capacidades de procesamiento de la información del cerebro imitando su estructura básica (Dayhoff, 1990). Consiste en una red de unidades o nodos de procesamiento simple interconectadas, que procesan información en paralelo. Esta característica permite a las RNA aprender patrones inmersos en los datos, al igual que el cerebro humano, en lugar de estar pre-programado.

Otra técnica para el análisis de patrones, es el clustering. El clustering o agrupamiento consiste en una técnica de aprendizaje automático sin supervisión. Es la clasificación no supervisada de patrones en grupos o clúster, entiéndase por patrones: observaciones, elementos de datos, o vectores de características. El clustering es útil para situaciones exploratorias de análisis de patrones, agrupamiento, toma de decisiones, minería de datos, recuperación de documentos, segmentación de imágenes (Jain, Murty, & .Flynn, 1999).

Existen diferentes algoritmos de agrupamiento útiles para el análisis de patrones en un conjunto de datos. Los Mapas Auto - Organizados de Kohonen, (Kohonen T. , Self-organized formation of topologically correct feature., 1988) del inglés SOM (Self Organized Maps), se ha convertido en uno de los algoritmos más utilizados en el ámbito de agrupamiento de datos, esto gracias a la baja dependencia al dominio del conocimiento y a los eficientes algoritmos de aprendizaje disponibles.

Estas estructuras resaltan por su capacidad de generar mapas topológicos a través de una arquitectura paralela y distribuida. Dichos mapas pueden ser vistos como una representación en bajas dimensiones de los datos de entrada, preservando las propiedades topológicas de la distribución (Kohonen T. , 1998).

Los mapas auto – organizados, han dado buenos resultados en tareas de agrupamiento de grandes conjuntos de datos, un ejemplo de esto es el caso de WEBSOM (Lagus, 1999); sin embargo poseen desventajas como la búsqueda de la unidad ganadora, el cual es un cálculo que se realiza durante la etapa de aprendizaje. Este cálculo implica que por cada patrón de entrada presentado a la red se realiza una comparación de proximidad con todas las unidades (neuronas) existentes, lo cual convierte al entrenamiento de la red en un proceso costoso (Cuadros-Vargas, 2004).

Situación

Realidad situacional, conflictos y nudos críticos

Actualmente, los sistemas de geo posicionamiento global han ganado un interés social significativo. Y es que con el avance de la tecnología, el GPS ha logrado incorporarse a la vida cotidiana con una gran variedad de servicios, por medio de teléfonos inteligentes, tabletas, sistemas de navegación en automóviles, etc.

Es entonces que la información generada por estos dispositivos, se convierte en un activo valioso, dentro de cualquier actividad humana. Sin embargo, a medida que la tecnología avanza, de igual forma crece la información, tanto en cantidad como complejidad, lo que hace necesario el uso de algoritmos que tengan como propósito, el análisis de grandes cantidades de información, con el objetivo de dar respuesta a problemas a fines.

Con la finalidad de obtener el conocimiento que se encuentra inmerso en estos datos, se plantea la investigación del algoritmo de mapas auto – organizados (SOM), con el propósito de aplicarlo en bases de datos científicas, que contienen información de trayectorias vehiculares para la detección patrones de comportamiento.

Los algoritmos de agrupamiento como herramienta para identificar patrones inmersos en un gran conjunto de datos, tiene varias aplicaciones en el ámbito de minería de datos: reconocimiento de imágenes, reconocimiento de voz, diagnóstico médico, segmentación de mercado, entre otros. Por lo cual se convierte en una necesidad entender cómo funcionan estos algoritmos y dónde se hace más eficiente su aplicación. Esto con el propósito de mejorar la gestión de toma de decisiones, en base a los resultados obtenidos de la aplicación del algoritmo, independientemente del área dónde se aplique.
Una de las principales causas que se pueden determinar del problema es el desconocimiento de los algoritmos de agrupamiento y de las herramientas que ayudan al campo de la minería de datos.
Otras causas que se pueden generar, una vez que se tiene una noción de estos algoritmos, es el costo computacional que realizan dichos algoritmos, lo cual demanda más recursos para el procesamiento de la información. Por

consiguiente obtendremos un agrupamiento de datos ineficiente, con altos márgenes de error. Lo que generaría falsos positivos en la interpretación de los resultados obtenidos.

La necesidad de organizar grandes cantidades de datos en grupos con significado para la identificación de patrones, ha hecho del agrupamiento una herramienta valiosa en el análisis de datos.

En el ámbito de trayectorias vehiculares puede ser de vital importancia, por ejemplo: para tomar decisiones sobre la planificación urbana, analizar congestión vehicular, comprender la migración de los animales, estudiar el comportamiento de los fenómenos naturales como los terremotos o sismos, etc.

Otras aplicaciones para el agrupamiento de datos son: la segmentación de imágenes, la minería de datos, reconocimiento de objetos, el procesamiento de lenguaje natural y segmentación de consumidores.

Estudios y teorias con base en la problemática

El Descubrimiento del conocimiento en bases de datos (Knowledge Discovery in Databases KDD) es un nuevo campo de la investigación que consiste en la extracción de información de alto nivel (conocimiento) (U., Piatetsky-Shapiro, Smyth, & Uthurusamy, 1996). Se ha convertido en un área de interés para los investigadores y profesionales de varios campos como por ejemplo: Inteligencia artificial, estadística, visualización, bases de datos, reconocimiento de patrones y computación paralela de alto rendimiento.

El KDD es un proceso que incluye varios pasos. Entre estos pasos tenemos: preparación y limpieza de los datos, selección y muestreo de los datos, pre procesamiento y transformación, minería de datos para extraer patrones y modelos, interpretación de la información extraída y finalmente la evaluación del conocimiento extraído (Fayyad, 1996).

En la actualidad encontramos diferentes dispositivos de comunicaciones, los cuales nos permiten obtener una gran cantidad de ubicaciones geográficas de determinados objetos (F. Giannotti, 2007). Es por eso que surge la necesidad de hacer uso de herramientas que permitan procesar dichos datos y poder generar conocimiento. Existen trabajos relacionados sobre patrones de trayectorias para la minería de datos (F. Giannotti, 2007) (Andrienko, 2011) (Andre Salvaro Furtado, 2012) (Fosca Giannotti, 2006) (Chih-Chieh Hung, 2015) (Salvatore Orlando, 2007).

La problemática básicamente es la búsqueda de patrones en bases de datos o modelos, lo cual puede ser de utilidad para el logro varios objetivos, como por ejemplo:

- Predicción (regresión y clasificación),
- Modelamiento descriptivo o generativo (clustering),
- Resumen de datos (generación de reportes), o
- Visualización de los datos o conocimiento extraído (soporte para tomas de decisiones o análisis exploratorio de datos)

Se han realizado investigaciones con respecto a algoritmos de aprendizaje automático y técnicas de optimización para dar solución a problemas complejos ya que, gracias a su capacidad de adaptación al entorno de información, han dado buenos resultados en distintas áreas, entre ellas la minería de datos (R. Sotolongo & Robles Aranda, 2013) y la robótica.

Redes Neuronales Artificiales (RNA)

Las Redes Neuronales Artificiales han sido utilizadas de forma satisfactoria en el ámbito de agrupamiento de datos, predicción, optimización; de tal forma que se ha convertido en una herramienta muy importante para la resolución de problemas de clasificación de patrones.

No existe un concepto general, si nos referimos a las redes neuronales artificiales, ya que existen diversos artículos, en donde el autor da su punto de vista con respecto a la misma. Es así que se detallan las siguientes definiciones:

- Una red neuronal consiste en un modelo de la computación, en paralelo, que está compuesta por unidades de procesamiento adaptivas con una alta interconexión entre dichas unidades (Hassoun, 1995).

- Son sistemas de procesamiento de información, que utilizan algunos principios fundamentales de la organización del cerebro humano (Lin, 1996).

- Emulación del cerebro humano, por medio de modelamientos matemáticos (Chen, 1998).

- Sistema de procesamiento de información, con características de funcionamiento comunes con las redes neuronales biológicas (Fausett, 1994).

- Sistema caracterizado por una red adaptiva, con técnicas de procesado de información en paralelo (Kung, 1993).

- En el ámbito de reconocimiento de patrones, las redes neuronales constituyen una extensión de los métodos estadísticos clásicos (Bishop, 1995).

Entre las características de las redes neuronales artificiales tenemos (Jain, Murty, & .Flynn, 1999):

    - Procesamiento de vectores numéricos, motivo por el cual los patrones deben ser representados con características cuantitativas.

- Cuentan con una arquitectura de procesamiento distribuido y paralelo.
- Tienen la capacidad de aprender los pesos de sus interconexiones adaptativamente, actuando como normalizadores de patrones y selectores de características.

Por lo general, una red neuronal artificial se divide en tres partes, denominadas capas (da Silva, 2017):

- **Capa de entrada.-** Esta capa es responsable de recibir información (datos), señales, características o mediciones del entorno externo. Estas entradas (muestras o patrones) normalmente se normalizan dentro de los valores límite producidos por las funciones de activación. Esta normalización resulta en una mejor precisión numérica para las operaciones matemáticas realizadas por la red.

- **Capas ocultas, intermedias o invisibles.-** Estas capas están compuestas de neuronas que son responsables de extraer patrones asociados con el proceso o sistema que se está analizando. Estas capas realizan la mayor parte del procesamiento interno desde una red.

- **Capa de salida.-** Esta capa también está compuesta de neuronas y, por lo tanto, es responsable de producir y presentar las salidas finales de la red, que resultan del procesamiento realizado por las neuronas en las capas anteriores.

Clasificación de las RNA según la arquitectura:

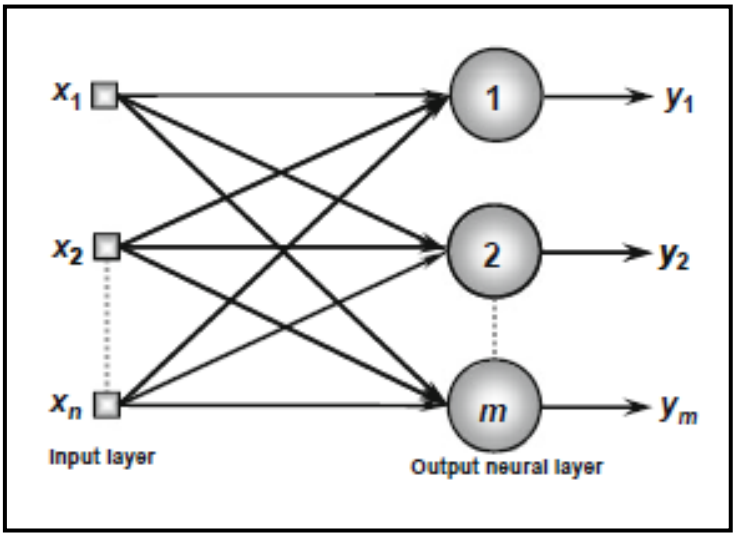
Las Redes Neuronales Artificiales, pueden representarse como un grafo dirigido con pesos, dónde los nodos son las neuronas artificiales y las aristas son las conexiones entre neuronas (Haykin, 1994).

Las principales arquitecturas de las redes neuronales artificiales, considerando la disposición de las neuronas, así como la forma en que están

interconectadas y cómo se componen sus capas, se pueden dividir de la siguiente manera:

- **Single-Layer Feedforward:** Este tipo de red posee sólo una capa de entrada y una sola capa neural, que es también la capa de salida. La información fluye siempre en una sola dirección (unidireccional), que es desde la capa de entrada a la capa de salida. Estas redes suelen emplearse en la clasificación de patrones y problemas de filtrado lineal (da Silva, 2017).

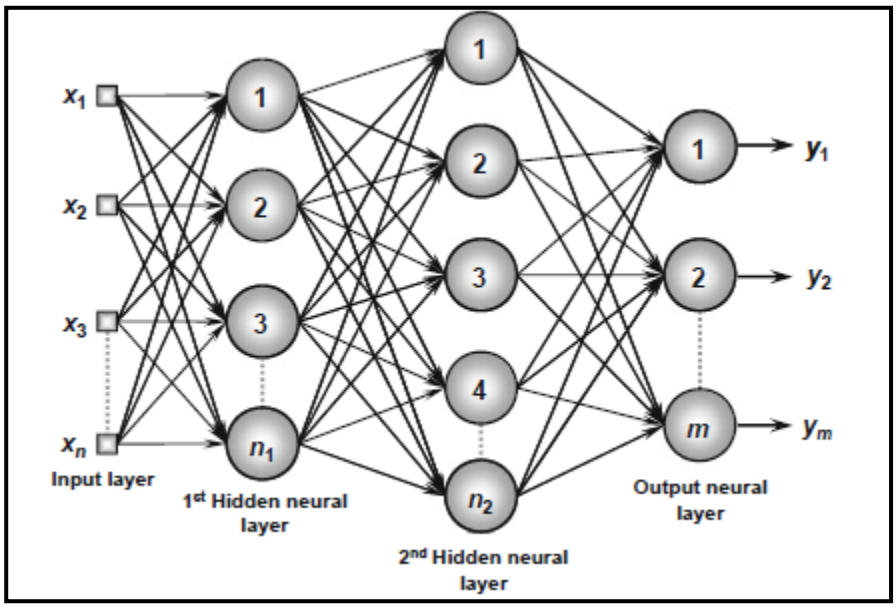
GRÁFICO 1 : Ejemplo de una Red Single-Layer Feedforward



Elaboración: da Silva, I.N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L.H.B., dos Reis Alves, S.F.
Fuente: (da Silva, 2017)

- **Multiple-Layer Feedforward:** A diferencia de la anterior, las redes feedforward con capas múltiples están compuestas de una o más capas neuronales ocultas. Se emplean en la solución de diversos problemas, como los relacionados con la aproximación de funciones, clasificación de patrones, identificación de sistemas, control de procesos, optimización, robótica, etc. Entre las principales redes que usan arquitecturas feedforward de múltiples capas se encuentran el Perceptrón multicapa (MLP) y la Función de Base Radial (RBF), cuyos algoritmos de aprendizaje utilizados en sus procesos de formación se basan respectivamente en la regla delta generalizada y la regla de competencia/delta.

GRÁFICO 2 : Ejemplo de una Red Multiple-Layer Feedforward

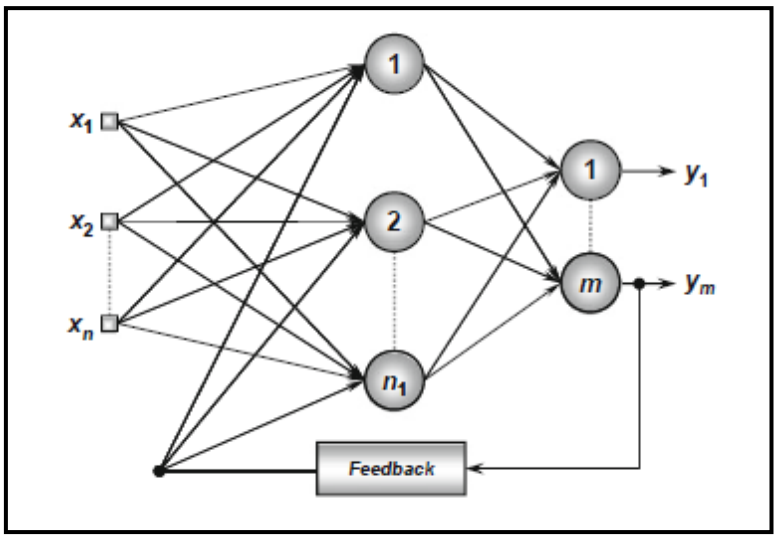


Elaboración: da Silva, I.N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L.H.B., dos Reis Alves, S.F.
Fuente: (da Silva, 2017)

- **Recurrentes o realimentadas (recurrent):** Son aquellas que admiten ciclos, es decir permite trazar caminos de una neurona entre sí (Jain, Murty, & .Flynn, 1999).

  En estas redes, las salidas de las neuronas se utilizan como entradas de retroalimentación para otras neuronas. La característica de realimentación califica estas redes para el procesamiento dinámico de la información, lo que significa que pueden emplearse en sistemas de variantes de tiempo, tales como predicción de series temporales, identificación y optimización de sistemas, control de procesos, etc. (da Silva, 2017).

GRÁFICO 3 : Ejemplo de una Red Recurrente



Elaboración: da Silva, I.N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L.H.B., dos Reis Alves, S.F.
Fuente: (da Silva, 2017)

Clasificación de las RNA según el aprendizaje:

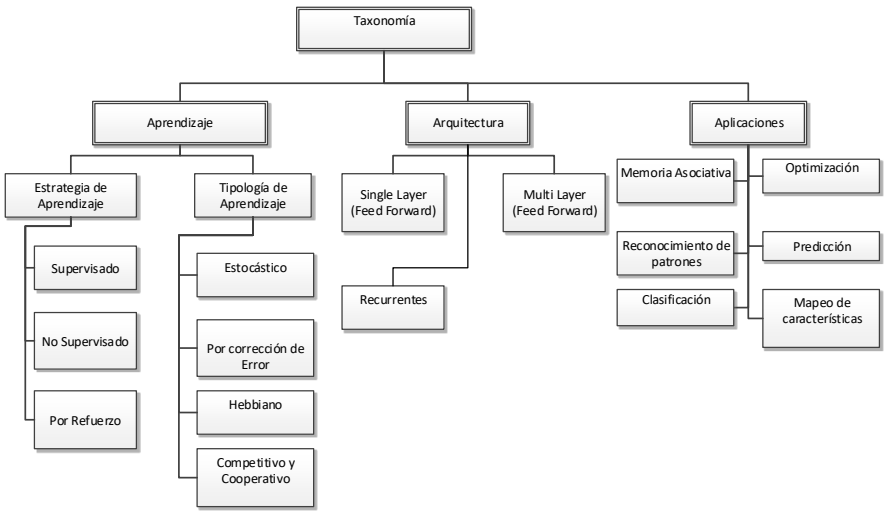
- **Aprendizaje supervisado:** La estrategia de aprendizaje supervisado consiste en disponer de las salidas deseadas para un conjunto dado de señales de entrada; En otras palabras, cada muestra de entrenamiento está compuesta por las señales de entrada y sus correspondientes salidas. En lo sucesivo, se requiere una tabla con datos de entrada / salida, también llamada tabla de atributo / valor, que representa el proceso y su comportamiento. Es a partir de esta información que las estructuras neurales formularán "hipótesis" sobre el sistema que se está aprendiendo (da Silva, 2017).

- **Aprendizaje no supervisado:** Al igual que el aprendizaje no supervisado, utiliza un conjunto de patrones de entrada; sin embargo no se establece una salida. Por lo tanto, la red necesita organizarse cuando existan particularidades entre los elementos que componen el conjunto de muestras completo, identificando subconjuntos que presentan similitudes. El algoritmo de aprendizaje ajusta los pesos y umbrales sinápticos de la red para reflejar estos grupos dentro de la propia red.

Alternativamente, el diseñador de red puede especificar (a priori) la cantidad máxima (da Silva, 2017).

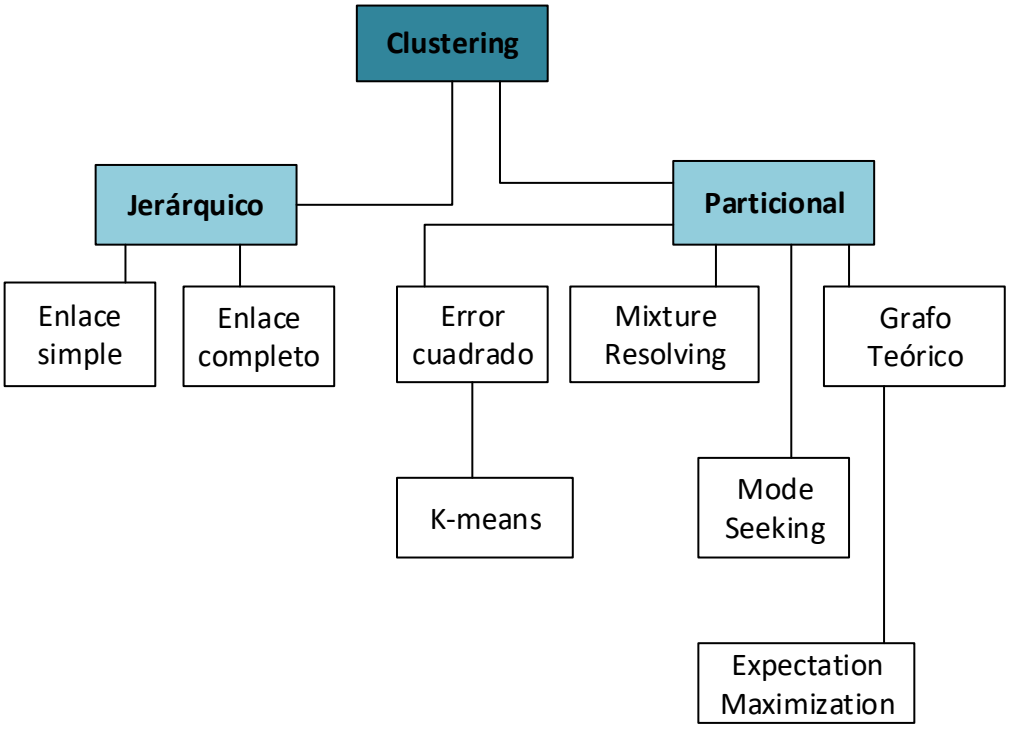
- **Aprendizaje por refuerzo:** Es considerado una variación de las técnicas de aprendizaje supervisado, ya que analizan continuamente la diferencia entre la respuesta producida por la red y la salida deseada correspondiente (Sutton & Barto, 1998). Los algoritmos que utilizan el aprendizaje de refuerzo ajustan los parámetros neuronales internos basándose en cualquier información cualitativa o cuantitativa recibida a través de la interacción con el sistema (entorno) que se mapea, utilizando esta información para evaluar el rendimiento de aprendizaje.
  El proceso de aprendizaje en red suele hacerse por ensayo y error porque la única respuesta disponible para una entrada dada es si fue satisfactoria o insatisfactoria. Si es satisfactorio, los pesos y umbrales sinápticos se incrementan gradualmente para reforzar (recompensar) esta condición de comportamiento involucrada con el sistema (da Silva, 2017).

GRÁFICO 4 : Taxonomía de los algoritmos de Redes Neuronales Artificiales



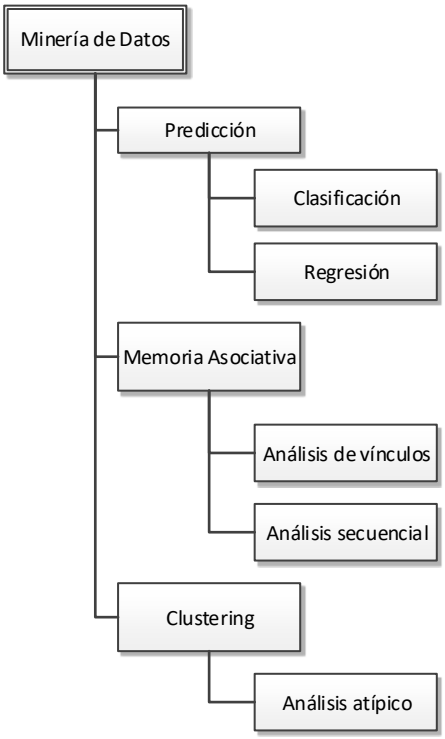
Elaboración: (da Silva, 2017)
Fuente: Artificial Neural Networks. A Practical Course

GRÁFICO 5 : Taxonomía enfocada a clustering



Elaboración: (Jain, Murty, & .Flynn, 1999)
Fuente: Data Clustering: A Review

GRÁFICO 6 : Taxonomía de los algoritmos en minería de datos



Elaboración: (Turban Efraim, 2011)
Fuente: Libro: Decision Support and Business Intelligence Systems

Método de aprendizaje	Algoritmo popular
Supervisado	Arboles de clasificación y regresión, ANN, SVM, Algoritmos Genéticos
Supervisado	Arboles de Decisión, ANN/MLP, SVM, Rough sets, Algoritmos Genéticos
Supervisado	Regresión Lineal y No Lineal, Arboles de Regresión, ANN/MLP, SVM
No Supervisado	Apriory, OneR, ZeroR, Eclat
No Supervisado	Expectation Maximization, Apriory Algorithm, Graph-based Matching
No Supervisado	Apriory Algorithm, FP-Growth technique
No Supervisado	K-means, ANN/SOM
No Supervisado	K-means, Expectation Maximization

Algoritmos de Agrupamiento

K-means

K-means es un método de partición, bien conocido. El resultado que arroja el algoritmo es un conjunto de K grupos, donde cada objeto del conjunto de datos pertenece a un grupo. En cada grupo puede haber un centroide o un grupo representativo. En el caso en que consideremos datos de valores reales, la media aritmética de los vectores de atributos para todos los objetos dentro de un grupo proporciona un representante apropiado; en otros casos pueden ser necesarios otros tipos de centroide (Singh, Malik, & Sharma, 2011) (Kaur, Sahiwal, & Kaur, 2012).

Clustering Jerárquico (Hierarchical Clustering)

El clustering jerárquico o agrupamiento jerárquico, organiza objetos en un dendrograma cuyas ramas son los clústeres deseados. El proceso de detección de racimos se denomina corte de árboles, corte de ramas o poda de ramas. El método de corte de árboles más común, al que nos referimos como el árbol "estático" cortado, define cada rama contigua debajo de un corte de altura fijo un grupo separado. La estructura de las alturas de unión de clústeres a menudo plantea un desafío a la definición de clúster. Aunque distintos grupos pueden ser reconocibles. (Langfelder, Zhang, & Horvath, 2007).

Mapas Auto Organizados (SOM)

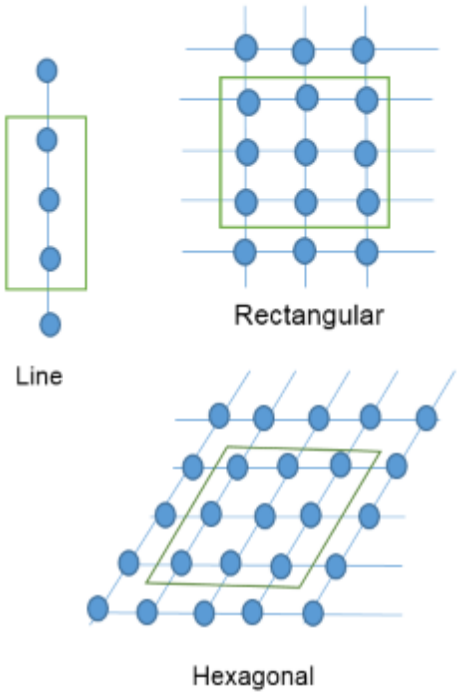
El algoritmo SOM fue descubierto por Teuvo Kohonen en Finlandia en el año 1982, consistía en un sistema con un comportamiento similar al del cerebro, con la capacidad para formar mapas de características de manera similar a como ocurre en el cerebro. En dicho mapa, hay neuronas que se organizan en muchas zonas, de forma tal que la información receptada del entorno a través de los órganos sensoriales se representa internamente en forma de mapas bidimensionales. Pertenecen al grupo de Redes Neuronales Artificiales (RNA), y corresponden a un tipo de aprendizaje no supervisado.

Arquitectura típica de un mapa SOM

Es una red de tipo unidireccional, está organizada en dos capas: la primera capa está formada por las neuronas de entrada mientras que la segunda capa consiste en un array de dos dimensiones a la cual denominaremos capa oculta. La conexión entre las neuronas de la primera capa con las neuronas de la segunda se denominan sinapsis (w), a las cuales se les etiquetara un peso que consta de tres índices i, j, k donde (i, j) indica la posición de la neurona en la capa y k, la componente o conexión con cierta neurona de entrada.

Los mapas auto-organizados poseen diferentes arquitecturas, entre las más utilizadas se encuentra la arquitectura lineal, la cual está formada por un arreglo de neuronas uni-dimensional, donde cada neurona posee dos vecinas directas, excepto las neuronas de los extremos que solo tienen una. También está la matriz rectangular. Aquí las neuronas pueden tener hasta cuatro vecinas directas. Y por último la hexagonal, donde las neuronas pueden tener hasta seis neuronas vecinas directas (Hasperué, 2005).

GRÁFICO 7 : Arquitecturas de los mapas auto - organizados



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: (Hasperué, 2005)

Características del algoritmo SOM

Son redes que utilizan un aprendizaje no supervisado competitivo. Cada neurona en la red utiliza como regla de propagación una distancia de su vector de pesos sinápticos al patrón de entrada. Otros conceptos importantes que intervienen en el proceso de este aprendizaje son los conceptos de neurona ganadora y vecindad de la misma. La neurona ganadora o Best

Matching Unit (BMU), es el vector de referencia o prototipo, que es cercana al patrón de entrada (Kohonen T. , Self-organization and associative memory., 1989) (Teuvo, 1990).

Algoritmo de entrenamiento

Un algoritmo de aprendizaje que describe el comportamiento de este tipo de red es el algoritmo de Kohonen, el cual consiste en lo siguiente:

1. Inicialización de los pesos w_{ij}.
2. Elección de un patrón entre el conjunto de patrones entrenamiento.
3. Para cada neurona del mapa, se calcula la distancia euclidiana entre el patrón de entrada x y el vector de pesos sinápticos:

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2$$

4. Evaluar la neurona ganadora (aquella cuya distancia es la menor de todas).
5. Actualizar los pesos sinápticos de la neurona ganadora y de sus vecinas según la regla:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)(x_i(t+1) - w_{ij}(t))$$

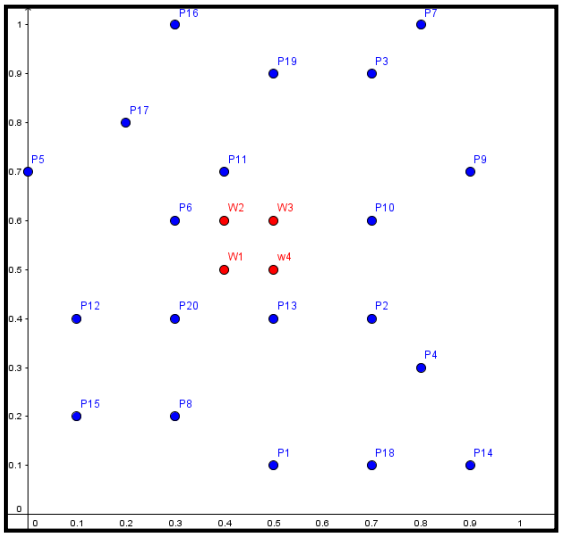
6. $\alpha(t)$ Es un factor llamado ritmo de aprendizaje que da cuenta de la importancia que la diferencia entre el patrón y los pesos que tiene el ajuste de los mismos a lo largo del proceso de aprendizaje.

Usualmente se fija un número de iteraciones antes de comenzar el aprendizaje. Si no se llegó al número de iteraciones establecida previamente, se vuelve al paso 2. Sobre este número de iteraciones necesario, se suelen tomar criterios como el número de neuronas en el mapa.

Ejemplo:

Vamos a utilizar una red con dos neuronas de entrada: coordenadas X e Y, los cuales serán los puntos a clasificar, y 4 neuronas de salida, motivo por el cual la red constituirá cuatro categorías de puntos. Los patrones de entrada corresponderán al intervalo [0,1]. En este ejemplo, no se establecerá ninguna zona de vecindad.

GRÁFICO 8 : Red con 2 neuronas de entrada y 4 de salida



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Carlos Cervantes, realizado en Geogebra 5

Seleccionamos los valores de los pesos, los cuales se ubican en el centro (puntos de color rojo) del plano XY:

W1 = [0.4, 0.5], W2 = [0.4, 0.6], W3 = [0.5, 0.6], W4 = [0.5, 0.5]

El valor de alfa será igual a: $\alpha(t) = 1/t$

Donde t es número de época o iteración.

El conjunto de patrones está conformado por 20 coordenadas XY:

P1 = [0.5, 0.1]	P2 = [0.7, 0.4]	P3 = [0.7, 0.9]	P4 = [0.8, 0.3]
P5 = [0, 0.7]	P6 = [0.3, 0.6]	P7 = [0.8, 1]	P8 = [0.3, 0.2]
P9 = [0.9, 0.7]	P10 = [0.7, 0.6]	P11 = [0.4, 0.7]	P12 = [0.1, 0.4]
P13 = [0.5, 0.4]	P14 = [0.9, 0.1]	P15 = [0.1, 0.2]	P16 = [0.3, 1]
P17 = [0.2, 0.8]	P18 = [0.7, 0.1]	P19 = [0.5, 0.9]	P20 = [0.3, 0.4]

Determinamos la distancia euclidiana para cada uno de los vectores de peso. El que posea la mínima distancia, será el patrón de entrada P:

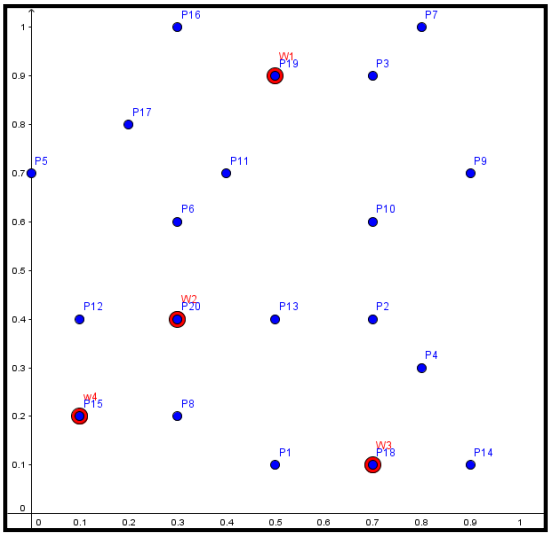
Primer patrón: P1 = [0.5, 0.1]

$$D_j = (p_x - w_{jx})^2 + (p_y - w_{jy})^2$$

$$D_1 = (0{,}5 - 0{,}4)^2 + (0{,}1 - 0{,}5)^2 = 0{,}17$$
$$D_2 = (0{,}5 - 0{,}4)^2 + (0{,}1 - 0{,}6)^2 = 0{,}26$$
$$D_3 = (0{,}5 - 0{,}5)^2 + (0{,}1 - 0{,}6)^2 = 0{,}25$$
$$D_4 = (0{,}5 - 0{,}5)^2 + (0{,}1 - 0{,}5)^2 = 0{,}16$$

Como podemos notar la menor distancia es 0,16, entonces el peso a actualizar será W4. Al finalizar la primera iteración realizando el algoritmo con los 20 patrones, el plano cartesiano que tenemos como resultado es el siguiente:

GRÁFICO 9 : Evolución de los pesos después de la primera iteración



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Carlos Cervantes, realizado en Geogebra 5

Etapas del entrenamiento de los mapas auto-organizados

- **Etapa de ordenamiento global o topológico.-** Durante esta etapa tiene lugar el ordenamiento topológico de los vectores de peso. Típicamente, esto requerirá hasta 1000 iteraciones del algoritmo SOM, y se debe prestar atención a la elección de los parámetros de la vecindad y de la tasa de aprendizaje (Bullinaria, 2004).

- **Etapa de convergencia.-** Durante esta etapa, el mapa de características está bien afinado y viene a proporcionar una cuantificación estadística precisa del espacio de entrada. Normalmente el número de iteraciones en esta fase será al menos 500 veces el número de neuronas en la red, y de nuevo los parámetros deben ser seleccionados cuidadosamente (Bullinaria, 2004).

Variantes de una Red SOM:

- **Growing Self-Organizing Maps.-** El Growing Self-Organizing Maps (GSOM) (Alahakoon, 1998), permite el crecimiento de la red, en forma dinámica similar al algoritmo IGG, inicialmente se tienen cuatro

neuronas conectadas formando un rectángulo, y nuevas unidades son insertadas en base a la unidad con mayor error acumulado. A diferencia de IGG, GSOM posee un método de inicialización de pesos, esto con el fin de reducir la probabilidad de generar mapas inapropiados.

- **Growing Hierarchical Self-Organizing Map.-** La idea clave del Growing Hierarchical Self- Organizing Map (GH-SOM) (Michael Dittenbach, 2000), es utilizar una estructura jerárquica de múltiples capas donde cada capa consta de un número de mapas independientes de auto-organización (SOM). Un SOM se utiliza en la primera capa de la jerarquía. Para cada unidad en este mapa se puede agregar un SOM a la siguiente capa de la jerarquía. Este principio se repite con la tercera y otras capas del GHSOM.

Métricas de calidad SOM

Error de cuantificación

El error de cuantificación (QE) se relaciona tradicionalmente con todas las formas de cuantificación vectorial y algoritmos de agrupamiento. Por lo tanto, esta medida no tiene en cuenta la topología del mapa y la alineación. El error de cuantificación se calcula determinando la distancia media de los vectores de muestra a los centroides de agrupamiento por los que están representados.

En el caso de la SOM, los centroides del clúster son los vectores prototipo. La medición de los errores de cuantificación puede extenderse de tal manera que funcione con conjuntos de datos que contienen valores faltantes. Para cualquier conjunto de datos dado, el error de cuantificación se puede reducir simplemente aumentando el número de nodos del mapa, porque entonces las muestras de datos se distribuyen más escasamente en el mapa. Debido a la compensación entre la cuantificación del vector y las propiedades de proyección de la SOM, el cambio del proceso de entrenamiento de tal manera que el QE se reduce conduce usualmente a la distorsión de la

topología del mapa (Pölzlbauer, 2004). Un mapa SOM con un error promedio bajo es más preciso, que un SOM con un error promedio alto.

$$\varepsilon_{qe} = \frac{1}{N}\sum_{j=1}^{N}\left\|x_j - m_{c(j)}\right\|^2$$

Error topográfico

El error topográfico es la más simple de las medidas de conservación de topología (Pölzlbauer, 2004). Este cálculo se realiza de la siguiente manera: Para todas las muestras de datos, se determinan las unidades de adaptación respectivas mejor y segunda mejor. Si éstos no son adyacentes en el enrejado del mapa, esto se considera un error.

El error total se normaliza a un rango de 0 a 1, donde 0 significa una preservación de topología perfecta. Normalmente, se devuelve un valor único que cuantifica esta propiedad. Sin embargo, es posible descomponer el error topográfico de tal manera que se puedan visualizar en un enrejado de mapa. Esto puede hacerse, por ejemplo, aumentando el error de una unidad cada vez que se selecciona como BMU por una muestra de datos, y la segunda BMU no es adyacente en el espacio de salida. El error topográfico se puede calcular para conjuntos de datos que contienen valores faltantes.

$$\varepsilon_t = \frac{1}{N}\sum_{k=1}^{N} u(x_k), \text{donde } u(x_k) \begin{cases} 1, & 1st\ and\ 2nd\ BMUs\ not\ adjacent \\ 0, & otherwise \end{cases}$$

En la implementación del algoritmo en lenguaje R, se especifican dos tipos de errores topográficos, dependiendo del valor del argumento type de la función topo.error (Kiviluoto, 1996) (Kohonen T. , Self-organizing maps, 2001):

1. **nodedist:** la distancia media, en términos de coordenadas (x, y) en el mapa, entre todos los pares de vectores de libro de códigos más similares.

2. **BMU:** la distancia media, en términos de coordenadas (x, y) en el mapa, entre la mejor unidad de coincidencia y la segunda mejor unidad de coincidencia, para todos los puntos de datos.

Ventajas del Algoritmo SOM:

Entre sus ventajas se pueden mencionar:

- Cuenta con un algoritmo relativamente simple, que brinda la facilidad de explicar resultados con datos no científicos.
- Se puede realizar el mapeo de nuevos datos, para el entrenamiento de un modelo con propósitos predictivos.
- Mantiene relación con otros algoritmos de agrupamiento, como por ejemplo los métodos de cuantificación vectorial.
- Permite identificar patrones de comportamiento en los datos, haciendo uso de todas las variables del modelo.

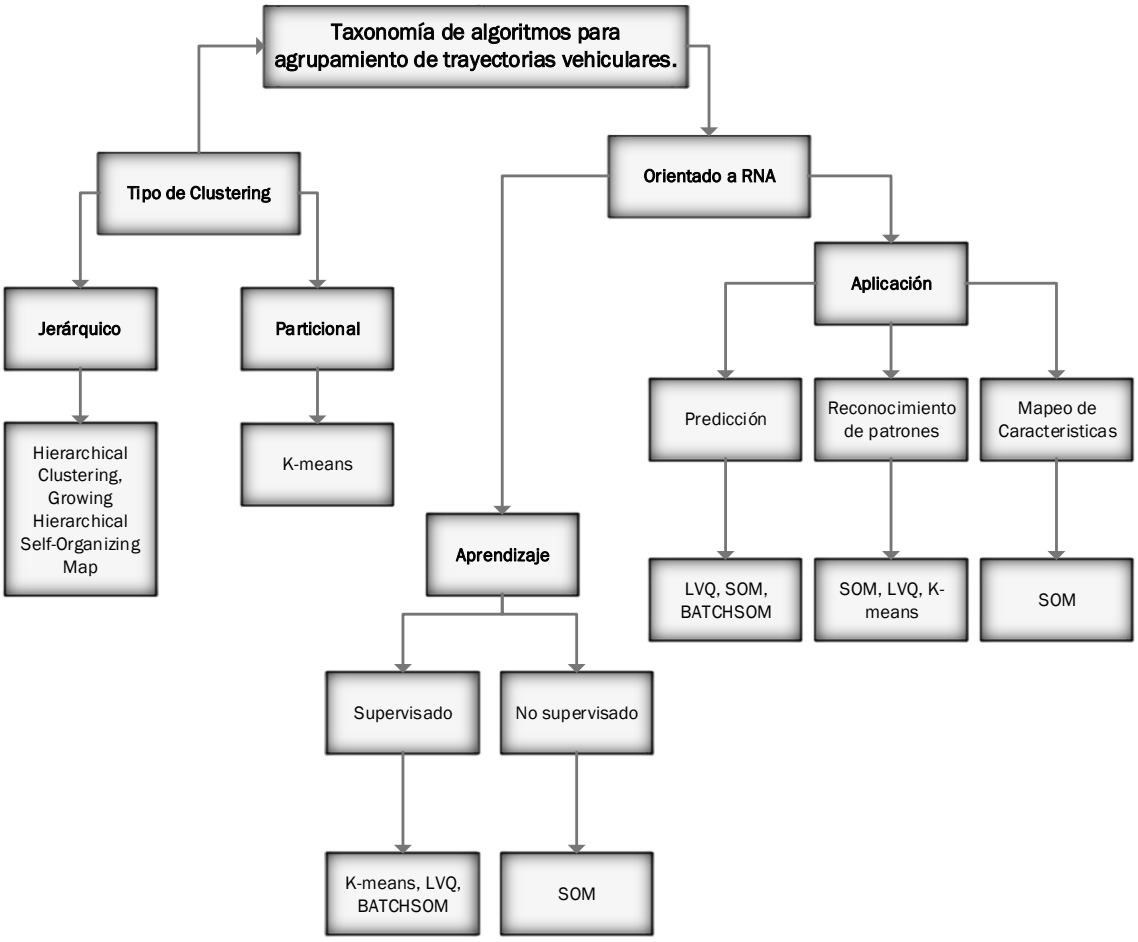
Limitaciones del Algoritmo SOM:

Entre sus desventajas se incluyen:

- Falta de capacidades de computación en paralelo para conjuntos de datos muy complejos, dado que el conjunto de datos de entrenamiento es iterativo.
- Demanda datos numéricos limpios.
- Hace dificultoso la representación de muchas variables en planos bidimensionales.
- Las redes con arquitecturas SOM, frecuentemente están condicionadas por activación de un solo ganador y representaciones estáticas.
- Lentitud de aprendizaje por el costo computacional del cálculo de distancia euclidiana.
- Predestina la topología de la red SOM.
- Carecen de la capacidad para dar respuesta a búsquedas específicas, como búsquedas por rangos o búsquedas de los k-vecinos más cercanos.

- Un problema con el algoritmo SOM es que la preservación de la topología del mapa no está garantizada, incluso si se realiza una gran cantidad de iteraciones (Martinetz & Schulten, 1994).

GRÁFICO 10 : Taxonomía de algoritmos para agrupamiento de trayectorias vehiculares



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Carlos Cervantes

Diferencias del algoritmo de Mapas Auto – Organizados con otros algoritmos

Diferencias entre SOM y K-Means

En K-Means los nodos (centroides) son independientes el uno del otro. El nodo ganador tiene la oportunidad de adaptar cada uno y sólo eso. En SOM los nodos (centroides) se colocan en una rejilla y por lo tanto cada nodo se

considera tener algunos vecinos, los nodos adyacentes o cerca de ella con respecto a su posición en la cuadrícula. Así que el nodo ganador no sólo se adapta sino que también provoca un cambio para sus vecinos.

Diferencias entre LVQ y SOM

Ambas algoritmos se basan en el principio de formación de mapas topológicos para establecer características comunes entre las informaciones (vectores) de entrada a la red, aunque difieren en las dimensiones de éstos, siendo de una sola dimensión en el caso de LVQ, y bidimensional, e incluso tridimensional, en la red SOM.

Aplicabilidad del algoritmo de Mapas Auto – Organizados

A continuación se detallan ciertos sistemas en los cuales se ha aplicado el algoritmo de mapas auto – organizados:

ViBlioSOM: Es una herramienta de visualización basada en el algoritmo de mapas auto - organizados que facilita la tarea de descubrir conocimiento inmerso en los datos. Dicha herramienta puede ser utilizada en cualquier campo del conocimiento, y es de utilidad para el análisis de correlación entre variables y datos complejos, para la clasificación de la información. Permite realizar filtros, de los grupos previamente formados y ahondar en el análisis de las variables que lo componen (G., M.V., & Carrillo H, 2002).

WEBSOM: Es una herramienta de navegación que utiliza un método exploratorio, para la recuperación de información de texto completo. En WEBSOM, documentos similares se asignan cerca uno del otro en el mapa, al igual que los libros en los estantes de una biblioteca bien organizada.

El WEBSOM es realmente aplicable a cualquier tipo de colección de documentos textuales. Es especialmente adecuado para tareas de exploración en las que los usuarios no conocen bien el dominio del tema, o tienen una idea limitada del contenido de la base de datos de texto completo que se está examinando (Kaski, Honkela, Lagus, & Kohonen, 1998).

LabSOM: Es un prototipo de software desarrollado por el Laboratorio de Dinámica no Lineal de la Facultad de Ciencias de la UNAM (Universidad Nacional Autónoma de México), mediante el cual el usuario puede realizar experimentos del algoritmo SOM, generando mapas en 2D y 3D, para la visualización de los datos de un determinado modelo (Jiménez-Andrade, Villaseñor-García, Escalera, Cruz-Ramírez, & Carrillo, 2007).

Definiciones

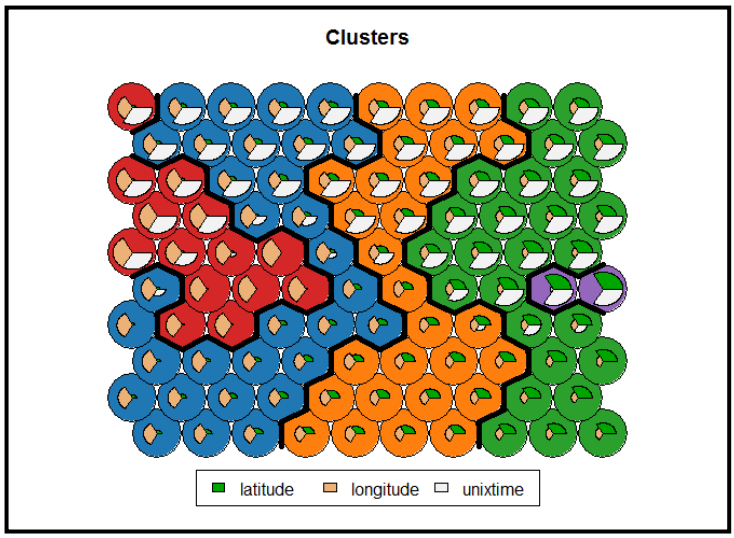
DEFINICIONES CONCEPTUALES

Definición 1: Auto – Organización.- Es una característica que posee la red neuronal para crear su propia organización, o representación de la información percibida, mediante una etapa de aprendizaje (Maren AJ, 1990). Consiste en un fenómeno observado en la naturaleza, mediante el cual se logra un orden global a partir de interacciones locales (Turing, 1952).

- **Ejemplo 1:** la evolución de la ubicación de los alumnos que acuden a un curso.
- **Ejemplo 2:** la auto-organización de las células del cerebro en grupos, según la información que albergan.

Definición 2: Topología.- La topología o arquitectura de una red neuronal consiste en la organización de las neuronas en la red, constituyendo capas o congregaciones de neuronas más o menos separadas de la entrada y salida de dicha red. De esta forma, los parámetros principales de la red son: el número de capas, el número de neuronas por capa, el nivel de conectividad y el tipo de conexiones entre neuronas (Haykin, 1994).

GRÁFICO 11 : Representación de un mapa SOM, con topología de 10x10



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Definición 3: Patrón.- Un patrón es una representación de un conjunto de trayectorias individuales que frecuentan la misma secuencia de lugares en intervalos de tiempo similares (F. Giannotti, 2007).

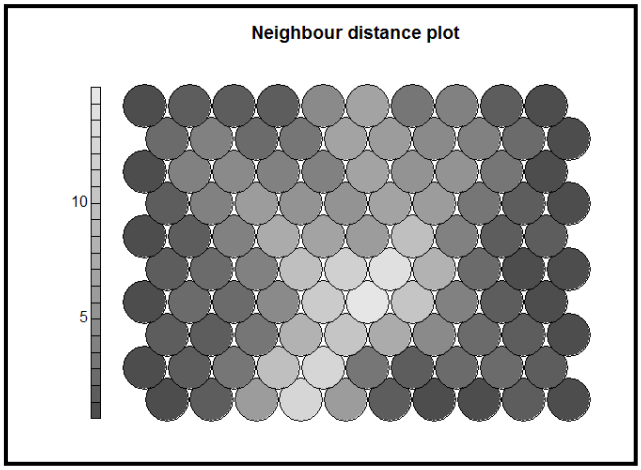
Definición 4: Centroide.- El centroide de un clúster, está definido como el punto equidistante de los objetos pertenecientes al clúster (Forgy, 1965) (McQueen, 1967).

Definición 5: Codebooks.- Un codebook describe la información sobre cada una de las variables de un conjunto de datos, además indica dónde y cómo se puede acceder a dicha información. Como mínimo el codebook debe incluir los siguientes ítems por cada variable (Trochin William, 2006):

- El nombre de la variable
- La descripción de la variable
- El formato de la variable
- Instrumento/método de recolección
- Datos recolectados
- Demandado o grupo
- Ubicación de la variable (en base de datos)
- Notas

Definición 6: U - Matrix (Matriz U).- La matriz U (matriz de distancia unificada) es una representación de un mapa de auto-organización donde la distancia eclídea entre los vectores de libro de códigos de neuronas vecinas se representa en una imagen en escala de grises. Esta imagen se utiliza para visualizar los datos en un espacio de alta dimensión utilizando la imagen 2D (A.Ultsch, 1990).

GRÁFICO 12 : Representación de la U-matrix

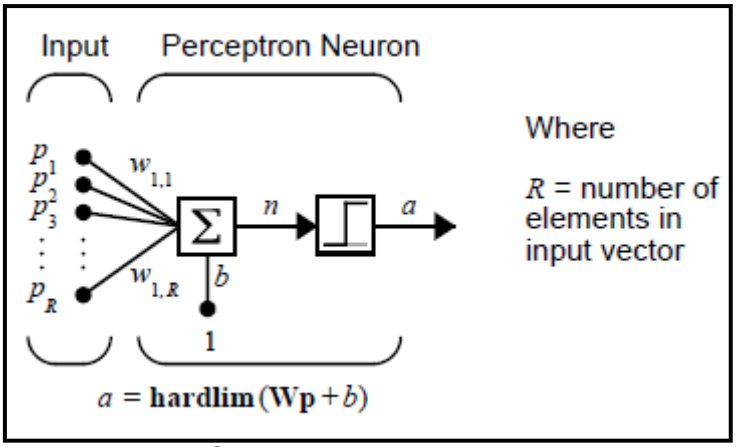


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Los colores claros representan vectores de libro de códigos de nodo reducidamente alejados y los colores más oscuros indican vectores de libro de códigos de nodos más ampliamente aislados.

Definición 7: Perceptrón.- Consiste en una sola neurona con pesos variables y un umbral. Un perceptrón imita a una neurona tomando la suma ponderada de sus entradas y enviando 1 a la salida, solo si dicha suma es mayor al umbral ajustable o 0 si ocurre lo contrario (Rich Elaine, 1994).

GRÁFICO 13 : El Perceptrón

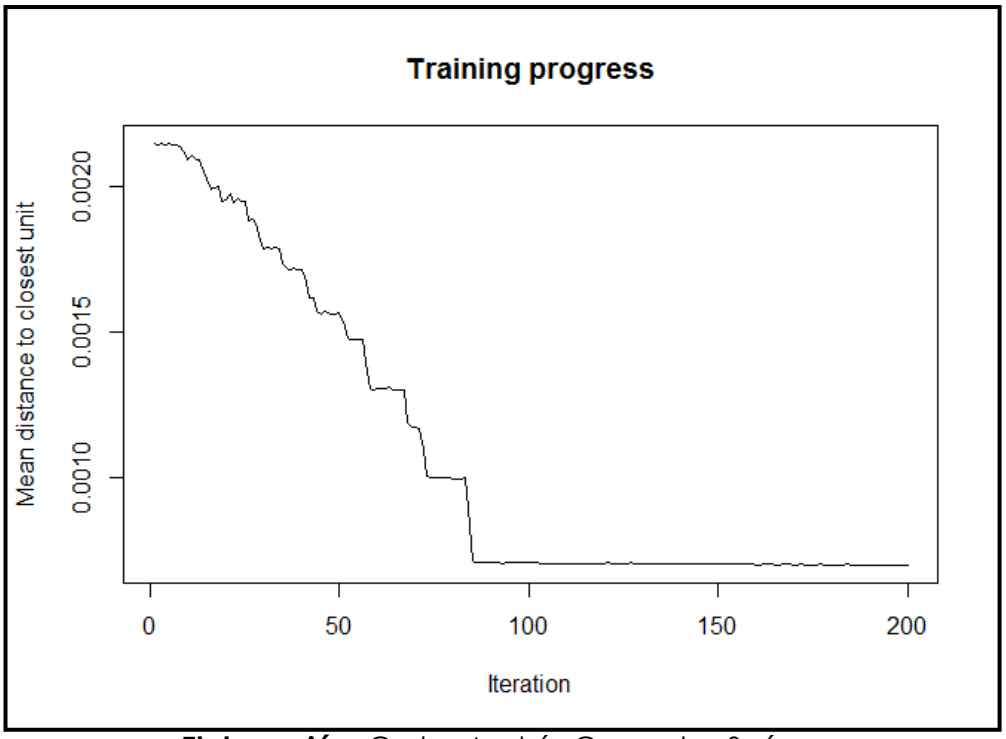


Elaboración: Demuth Howard, Beale Mark
Fuente: (Demuth Howard, 1992)

Definición 8: Aprendizaje.- La capacidad de aprender es una propiedad fundamental de la inteligencia. El proceso del aprendizaje, desde el contexto

de Redes Neuronales Artificiales, consiste en la actualización de la arquitectura de la red y de los pesos de conexión. (Haykin, 1994) (Bishop, 1995) (Jain A. K., 1996).

GRÁFICO 14 : Progreso de entrenamiento de un mapa SOM



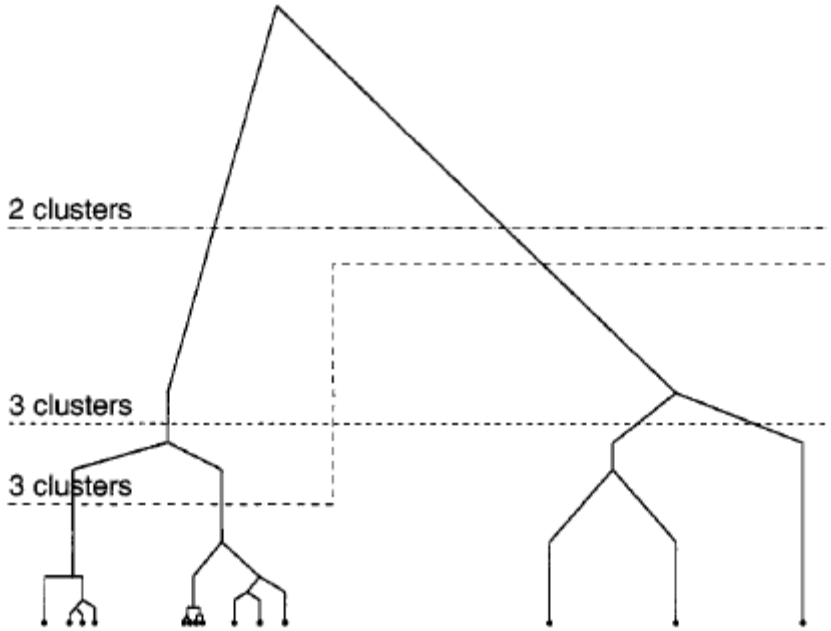
Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

En este gráfico se puede visualizar, el progreso de entrenamiento de un mapa SOM que inicia con un facto de aprendizaje de 0.09.

Definición 9: Dendrograma.- Un dendrograma es un tipo de representación gráfica, acíclico-binario arraigado con los siguientes tipos de vértices (Mesa & Restrepo, 2008):

1. Vértices de grado 1, llamados objetos.
2. Vértices de grado 3, llamados nodos.
3. Sólo un vértice de grado 2, llamado nodo raíz.

GRÁFICO 15 : Un Dendrograma



Elaboración: Juha Vesanto y Esa Alhoniemi
Fuente: (Vesanto & Alhoniemi, 2000)

Definición 10: Similitud de trayectorias: Existe similitud en dos trayectorias, cuando coinciden aproximadamente en sus puntos de origen y destino o si coinciden en algunas partes de las trayectorias (Swaminathan Sankararaman Pankaj K. Agarwal Thomas Molhave, 2013).

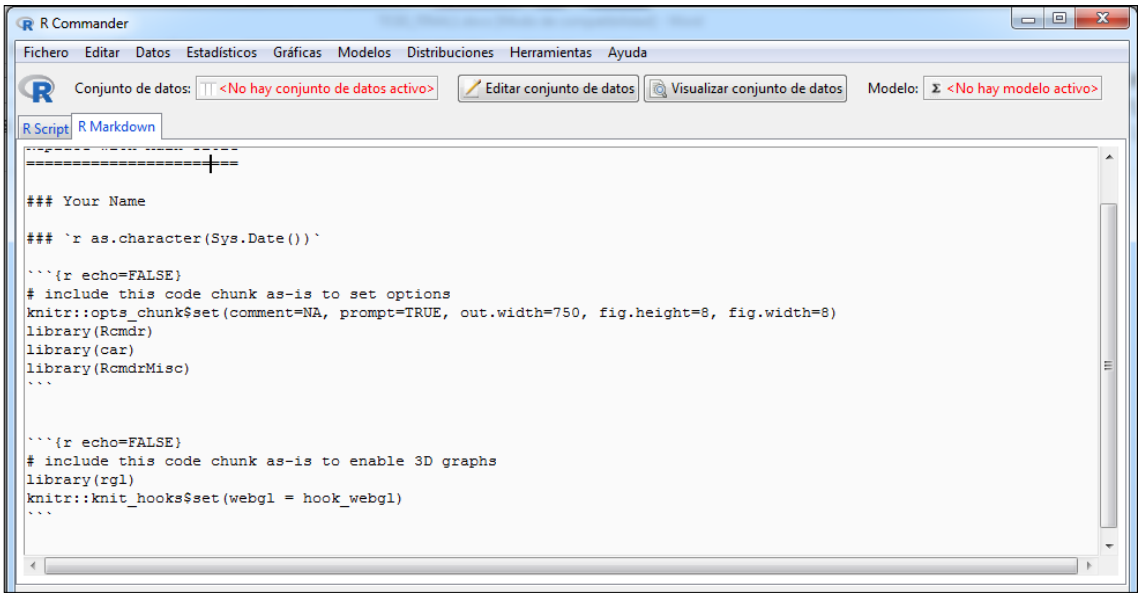
PROCESAMIENTO Y ANÁLISIS

El procesamiento de los datos se realizó por medio del gestor de base de datos POSTGRESQL. En este gestor de base de datos, se cargó un archivo backup, el cual contenía las tres bases de datos: california, plt, t-drive. El análisis y el cálculo de los resúmenes estadísticos para los datos de las trayectorias vehiculares se realizarán con R Commander.

R Commander:

R Commander proporciona una interfaz gráfica de usuario o GUI al entorno estadístico abierto R:

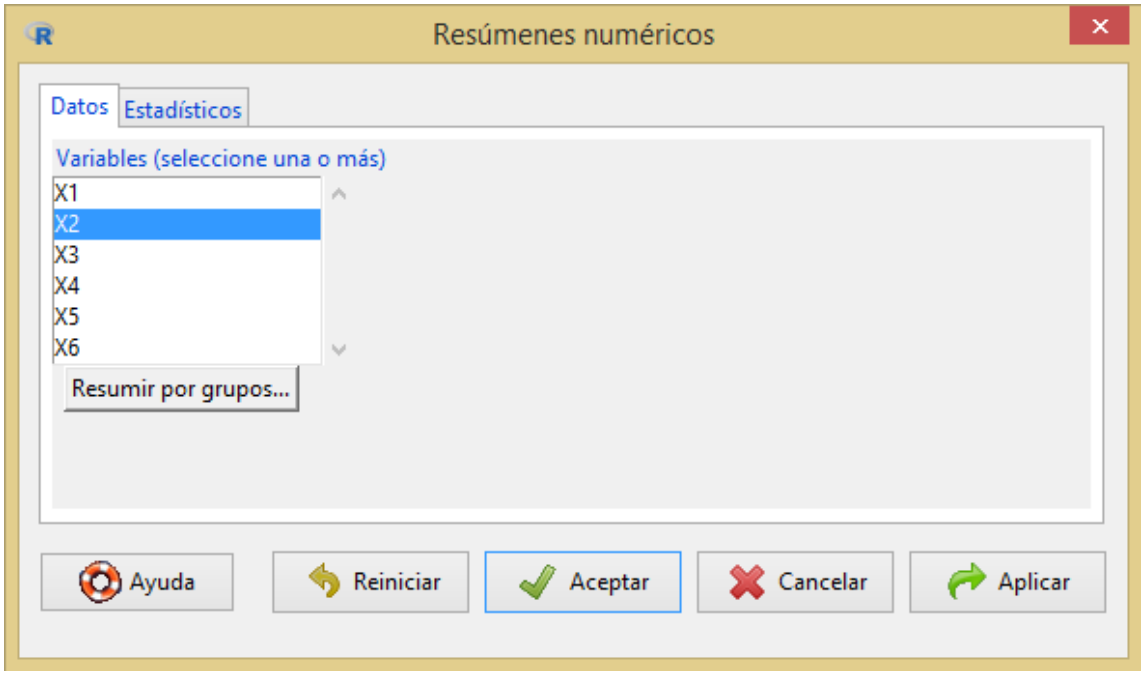
GRÁFICO 16 : GUI R Commander



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

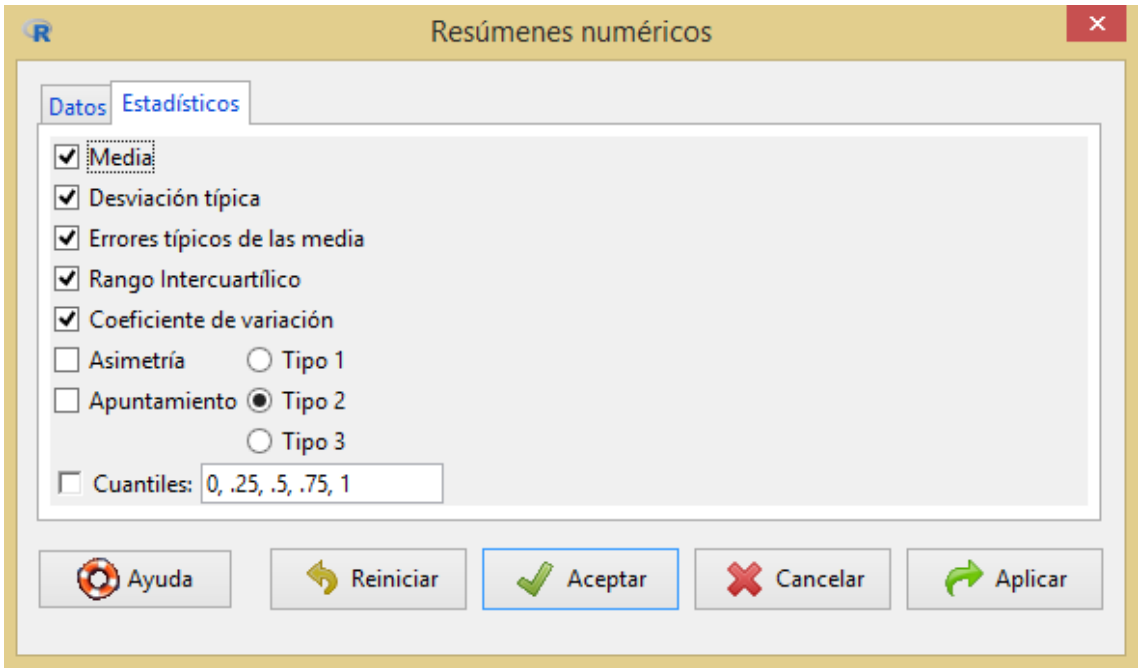
Para calcular la media, desviación estándar, error típico en la media y coeficiente de variación, nos ubicamos en la pestaña *Estadísticos-> Resúmenes->Resúmenes Numéricos*.

GRÁFICO 17 : R Commander: Resúmenes numéricos, pestaña Datos



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

GRÁFICO 18 : R Commander: Resúmenes numéricos, pestaña Estadísticos



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Contraste de Hipótesis

Las ejecuciones del algoritmo se realizaron en el IDE RStudio, el cual ofrece las herramientas necesarias para trabajar con el lenguaje de programación R, y por medio de la lógica programación, se procedió a almacenar los resultados obtenidos en un data.frame, el cual es un tipo de dato en R el cual tiene ciertas restricciones (Venables, Smith, & R-Team, 2016):

- Los componentes deben ser vectores (numéricos, de carácter o lógicos), factores, matrices numéricas, listas u otros marcos de datos.
- Matrices, listas y marcos de datos proporcionan tantas variables al nuevo marco de datos como tienen columnas, elementos o variables, respectivamente.
- Los vectores numéricos, los lógicos y los factores se incluyen como tal y, por defecto, los vectores de carácter se convierten en factores, cuyos niveles son los valores únicos que aparecen en el vector.
- Las estructuras vectoriales que aparecen como variables del marco de datos deben tener la misma longitud, y todas las estructuras de las matrices deben tener el mismo tamaño de fila.

Ejemplo:

```
n = c(2, 3, 5)
s = c("aa", "bb", "cc")
b = c(TRUE, FALSE, TRUE)
df = data.frame(n, s, b)      # df es un data.frame
```

Por medio de este data.frame, se almacenaron los resúmenes de las ejecuciones del algoritmo en una hoja de datos con seis valores:

- X1: Tiempo de carga de la data algoritmo
- X2: Tiempo de ejecución del algoritmo
- X3: Tiempo del proceso de clustering
- X4: Error topográfico - distancia entre nodos
- X5: Error topográfico - distancia entre BMU
- X6: Error de Cuantificación

Para conocer la exactitud y rendimiento del algoritmo sobre datos de trayectorias, se realizaron 100 ejecuciones del mismo. Esto con el propósito de tener un conjunto de información representativo, para poder evaluar la tendencia del algoritmo en base a las métricas.

A continuación se detalla el proceso de test de hipótesis, el cual fue realizado con las variables X2, X4, X5 y X6. Dichas variables, corresponden a las métricas del algoritmo SOM. El test de hipótesis se realiza para cada una de las variables, por cada base (california, plt, t-drive).

Base california

Combinación SOM-HC

X2: Tiempo de ejecución del algoritmo

Vamos a verificar que el tiempo de ejecución del algoritmo SOM-HC es diferente a 1.30, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.289066**
SD: **0.01892884**
SE (Mean): **0.001892884**
CV: **0.01468415**

Calculamos intervalo de confianza:
Con $\mu \neq 1.3$
data: X2
t = -5.7764, df = 99, p-value = 8.833e-08
alternative hypothesis: true mean is not equal to 1.3
95 percent confidence interval:
1.285310 1.292822

El tiempo de ejecución del algoritmo SOM-HC, es diferente de 1.30 de manera significativa dado que p-value es < a 0.05. Ahora vamos a comprobar que el tiempo de ejecución del algoritmo sea mayor 1.30.

Con μ > 1.3
data: X2
t = -5.7764, df = 99, p-value = 1
alternative hypothesis: true mean is greater than 1.3
95 percent confidence interval:
1.285923 Inf

El tiempo del algoritmo no es mayor a 1.30, de manera no significativa al 95% de confianza. El intervalo de confianza al 95% para μ es [1.285310, 1.292822] que no incluye al valor de 1.3. Según esto podemos decir que μ es menor.

X4: Error topográfico - distancia entre nodos

Se desea verificar que la distancia entre nodos en la ejecución del algoritmo SOM-HC es diferente a 1.20, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.226232**
SD: **0.06240295**
SE (Mean): **0.006240295**
CV: **0.05089**

Calculamos intervalo de confianza:
Con μ ≠ 1.20
data: X4
t = 4.2036, df = 99, p-value = 5.765e-05
alternative hypothesis: true mean is not equal to 1.2
95 percent confidence interval:
1.213850 1.238614

La distancia entre nodos que da como resultado la ejecución del algoritmo SOM-HC, es diferente de 1.20 de manera significativa dado que 5.765e-05 es < a 0.05. Ahora deseamos verificar si la distancia entre nodos es menor a 1.20, tras ejecutar el algoritmo SOM-HC.

Con μ < 1.20
data: X4
t = 4.2036, df = 99, p-value = 1
alternative hypothesis: true mean is less than 1.2
95 percent confidence interval:
-Inf 1.236593

La distancia entre nodos es no es menor que 1.2, de manera no significativa al 95%. El intervalo de confianza para μ es [1.213850, 1.238614]

X5: Error topográfico - distancia entre BMU

Se desea verificar que la distancia entre nodos en la ejecución del algoritmo SOM-HC es diferente a 1.20, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.193711**
SD: **0.07447633**
SE (Mean): **0.007447633**
CV: **0.06239058**

Calculamos intervalo de confianza:
Con μ ≠ 1.20
data: X5
t = -0.84441, df = 99, p-value = 0.4005
alternative hypothesis: true mean is not equal to 1.2
95 percent confidence interval:
1.178933 1.208489

La distancia entre mejor unidad ganadora que da como resultado la ejecución del algoritmo SOM-HC, no es diferente a 1.20 de manera significativa dado que 0.4005 no es < a 0.05. Ahora deseamos verificar si la

distancia entre mejor unidad ganadora es mayor que 1.20, tras ejecutar el algoritmo SOM-HC.

Con $\mu > 1.20$
data: X5
t = -0.84441, df = 99, p-value = 0.7998
alternative hypothesis: true mean is greater than 1.2
95 percent confidence interval:
1.181345 Inf

La distancia entre mejor unidad ganadora que da como resultado la ejecución del algoritmo SOM-HC, es menor o igual a 1.20 a un 95% de confianza, de manera no significativa (t = -0.84441, df = 99, p-value = 0.7998).

X6: Error de Cuantificación

Vamos a verificar que el error de cuantificación generado por el algoritmo SOM-HC es diferente a 0.01, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean:	**0.01213804**
SD:	**0.0005983356**
SE (Mean):	**5.983356e-05**
CV:	**0.04929424**

Calculamos intervalo de confianza:

Con $\mu \neq 0.01$
data: X6
t = 35.733, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.01
95 percent confidence interval:
0.01201932 0.01225677

El error de cuantificación que da como resultado de la ejecución del algoritmo SOM-HC, no es diferente a 0.01 de manera significativa dado que 2.2e-16 es < a 0.05. Ahora deseamos verificar el error de cuantificación es mayor que 0.01, tras ejecutar el algoritmo SOM-HC.

Con $\mu > 0.01$
data: X6
t = 35.733, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is greater than 0.01
95 percent confidence interval:
0.0120387 Inf

El error de cuantificación, no es mayor que 0.01 de manera significativa a un 95% de confianza (2.2e-16 es < a 0.05).

SOM-Kmeans

X2: Tiempo de ejecución del algoritmo

Vamos a verificar que el tiempo de ejecución del algoritmo SOM-Kmeans es diferente a 1.40, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.338865**
SD: **0.03770922**
SE (Mean): **0.003770922**
CV: **0.02816507**

Calculamos intervalo de confianza:
Con $\mu \neq 1.40$
data: X2
t = -16.212, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 1.4

95 percent confidence interval:
1.331383 1.346347

El tiempo de ejecución del algoritmo SOM-Kmeans, es diferente de 1.40 de manera significativa dado que p-value es < a 0.05. Ahora vamos a comprobar que el tiempo de ejecución del algoritmo sea mayor 1.40.

Con $\mu > 1.40$
data: X2
t = -16.212, df = 99, p-value = 1
alternative hypothesis: true mean is greater than 1.4
95 percent confidence interval:
1.332604 Inf

El tiempo del algoritmo no es mayor a 1.40, de manera no significativa al 95% de confianza. El intervalo de confianza al 95% para μ es [1.331383, 1.346347] que no incluye al valor de 1.4. Según esto podemos decir que μ es menor. Se corrobora que SOM-HC en comparación de tiempo con SOM-Kmeans, el tiempo es menor.

X4: Error topográfico - distancia entre nodos

Se desea verificar que la distancia entre nodos en la ejecución del algoritmo SOM-Kmeans es diferente a 1.20, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.223377**
SD: **0.06426835**
SE (Mean): **0.006426835**
CV: **0.05253357**

Calculamos intervalo de confianza:
Con $\mu \neq 1.20$
data: X4

t = 3.6374, df = 99, p-value = 0.0004395
alternative hypothesis: true mean is not equal to 1.2
95 percent confidence interval:
1.210625 1.236129

La distancia entre nodos que da como resultado la ejecución del algoritmo SOM-Kmeans, es diferente de 1.20 de manera significativa dado que 0.0004395 es < a 0.05. Ahora deseamos verificar si la distancia entre nodos es mayor a 1.23, tras ejecutar el algoritmo SOM-Kmeans.

Con $\mu > 1.23$
data: X4
t = -1.0306, df = 99, p-value = 0.8474
alternative hypothesis: true mean is greater than 1.23
95 percent confidence interval:
1.212706 Inf

La distancia entre nodos es menor o igual a 1.23, de forma no significativa (t = -1.030, df = 99, p-value = 0.8474) al 95% de confianza.

X5: Error topográfico - distancia entre BMU

Se desea verificar que la distancia entre nodos, es diferente a 1.20, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.204019**
SD: **0.07511682**
SE (Mean): **0.007511682**
CV: **0.06238841**

Calculamos intervalo de confianza:
Con $\mu \neq 1.20$
data: X5
t = 0.53501, df = 99, p-value = 0.5938

alternative hypothesis: true mean is not equal to 1.2
95 percent confidence interval:
1.189114 1.218924

La distancia entre mejor unidad ganadora, no es diferente a 1.20 de manera significativa dado que 0.5938 no es < a 0.05. Ahora deseamos verificar si la distancia entre mejor unidad ganadora es mayor que 1.20.

Con $\mu > 1.20$
data: X5
t = 0.53501, df = 99, p-value = 0.2969
alternative hypothesis: true mean is greater than 1.2
95 percent confidence interval:
1.191547 Inf

La distancia entre mejor unidad ganadora, es mayor o igual a 1.20 a un 95% de confianza, de manera significativa (t = 0.53501, df = 99, p-value = 0.2969).

X6: Error de Cuantificación

Se verifica que el error de cuantificación generado por el algoritmo SOM-Kmeans es diferente a 0.01, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **0.0120596**
SD: **0.0006661805**
SE (Mean): **6.661805e-05**
CV: **0.05524066**

Calculamos intervalo de confianza:
Con $\mu \neq 0.01$
data: X6
t = 30.917, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.01

95 percent confidence interval:
0.01192742 0.01219179

El error de cuantificación, no es diferente a 0.01 de manera significativa dado que 2.2e-16 es < a 0.05. Ahora deseamos verificar el error de cuantificación es mayor que 0.01, tras ejecutar el algoritmo SOM-Kmeans.

Con $\mu > 0.01$
data: X6
t = 30.917, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is greater than 0.01
95 percent confidence interval:
0.01194899 Inf

El error de cuantificación, no es mayor que 0.01 de manera significativa a un 95% de confianza (2.2e-16 es < a 0.05).

Base t-drive

SOM-HC

X2: Tiempo de ejecución del algoritmo

Vamos a verificar que el tiempo de ejecución del algoritmo SOM-HC es diferente a 1.30, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.294729**
SD: **0.03236434**
SE (Mean): **0.003236434**
CV: **0.024997**

Calculamos intervalo de confianza:
Con $\mu \neq 1.30$

data: X2
t = -1.6287, df = 99, p-value = 0.1066
alternative hypothesis: true mean is not equal to 1.3
95 percent confidence interval:
1.288307 1.301151

El tiempo de ejecución del algoritmo SOM-HC, no es diferente a 1.30 de manera significativa dado que 0.1066 no es < a 0.05. Ahora vamos a comprobar que el tiempo de ejecución del algoritmo sea mayor 1.30.

Con $\mu > 1.30$
data: X2
t = -1.6287, df = 99, p-value = 0.9467
alternative hypothesis: true mean is greater than 1.3
95 percent confidence interval:
1.289355 Inf

El tiempo del algoritmo no es mayor a 1.30, de manera no significativa al 95% de confianza. El intervalo de confianza al 95% para μ es [1.288307 1.301151].

X4: Error topográfico - distancia entre nodos

Se desea verificar que la distancia entre nodos en la ejecución del algoritmo SOM-HC es diferente a 1.80, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.703689**
SD: **0.1831418**
SE (Mean): **0.01831418**
CV: **0.1074972**

Calculamos intervalo de confianza:
Con $\mu \neq 1.80$
data: X4

t = -5.2588, df = 99, p-value = 8.381e-07
alternative hypothesis: true mean is not equal to 1.8
95 percent confidence interval:
1.667349 1.740028

La distancia entre nodos, es diferente de 1.80 de manera significativa dado que 8.381e-07 es < a 0.05. Ahora deseamos verificar si la distancia entre nodos es mayor a 1.80, tras ejecutar el algoritmo SOM-HC.

Con $\mu > 1.80$
data: X4
t = -5.2588, df = 99, p-value = 1
alternative hypothesis: true mean is greater than 1.8
95 percent confidence interval:
1.67328 Inf

La distancia entre nodos no es mayor que 1.80, de manera no significativa (t = -5.2588, df = 99, p-value = 1), al 95% de confianza. El intervalo de confianza es [1.667349, 1.740028].

X5: Error topográfico - distancia entre BMU

Se desea verificar que la distancia entre nodos, es diferente a 1.60, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.542065**
SD: **0.1700529**
SE (Mean): **0.01700529**
CV: **0.1102761**

Calculamos intervalo de confianza:
Con $\mu \neq 1.60$
data: X5
t = -3.4069, df = 99, p-value = 0.000951

alternative hypothesis: true mean is not equal to 1.6
95 percent confidence interval:
1.508323 1.575807

La distancia entre mejor unidad ganadora, es diferente a 1.60 de manera significativa dado que 0.000951 es < a 0.05. Ahora deseamos verificar si la distancia entre mejor unidad ganadora es mayor que 1.60.

Con μ > 1.60
data: X5
t = -3.4069, df = 99, p-value = 0.9995
alternative hypothesis: true mean is greater than 1.6
95 percent confidence interval:
1.51383 Inf

La distancia entre mejor unidad ganadora, no es mayor a 1.60 a un 95% de confianza, de manera no significativa (t = -3.4069, df = 99, p-value = 0.9995).

X6: Error de Cuantificación

Se verifica que el error de cuantificación generado por el algoritmo SOM-HC es diferente a 0.02, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **0.0224257**
SD: **0.00158631**
SE (Mean): **0.000158631**
CV: **0.07073625**

Calculamos intervalo de confianza:
Con μ ≠ 0.02
data: X6
t = 15.291, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.02

95 percent confidence interval:
0.02211094 0.02274046

El error de cuantificación, no es diferente a 0.02 de manera significativa dado que 2.2e-16 es < a 0.05. Ahora deseamos verificar el error de cuantificación es mayor que 0.02, tras ejecutar el algoritmo SOM-HC.

Con $\mu > 0.02$
data: X6
t = 15.291, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is greater than 0.02
95 percent confidence interval:
0.02216231 Inf

El error de cuantificación, no es mayor que 0.01 de manera significativa a un 95% de confianza (2.2e-16 es < a 0.05).

SOM-Kmeans

X2: Tiempo de ejecución del algoritmo

Vamos a verificar que el tiempo de ejecución del algoritmo SOM-Kmeans es diferente a 1.50, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean:	**1.401875**
SD:	**0.7968189**
SE (Mean):	**0.07968189**
CV:	**0.5683952**

Calculamos intervalo de confianza:
Con $\mu \neq 1.50$
data: X2

t = -1.2315, df = 99, p-value = 0.2211
alternative hypothesis: true mean is not equal to 1.5
95 percent confidence interval:
1.243769 1.559981

El tiempo de ejecución del algoritmo SOM-Kmeans, no es diferente a 1.50 de manera significativa dado que p-value no es < a 0.05. Ahora vamos a comprobar que el tiempo de ejecución del algoritmo sea mayor 1.60.

Con $\mu > 1.60$
data: X2
t = -2.4865, df = 99, p-value = 0.007288
alternative hypothesis: true mean is less than 1.6
95 percent confidence interval:
-Inf 1.534178

El tiempo de ejecución del algoritmo SOM-Kmeans, no es mayor a 1.60 de manera significativa (0.007288 es < a 0.05).

X4: Error topográfico - distancia entre nodos

Se desea verificar que la distancia entre nodos en la ejecución del algoritmo SOM-Kmeans es diferente a 1.80, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.766572**
SD: **0.1676493**
SE (Mean): **0.01676493**
CV: **0.09490095**

Calculamos intervalo de confianza:
Con $\mu \neq 1.80$
data: X4
t = -1.9939, df = 99, p-value = 0.04891
alternative hypothesis: true mean is not equal to 1.8

95 percent confidence interval:
1.733307 1.799837

La distancia entre nodos que da como resultado la ejecución del algoritmo SOM-Kmeans, es diferente de 1.80 de manera significativa dado que 0.04891 es < a 0.05. Ahora deseamos verificar si la distancia entre nodos es mayor a 1.80, tras ejecutar el algoritmo SOM-Kmeans.

Con $\mu > 1.80$
data: X4
t = -1.9939, df = 99, p-value = 0.9755
alternative hypothesis: true mean is greater than 1.8
95 percent confidence interval:
1.738735 Inf

La distancia entre nodos es menor a 1.80, de forma no significativa (t = -1.9939, df = 99, p-value = 0.9755) al 95% de confianza.

X5: Error topográfico - distancia entre BMU

Se desea verificar que la distancia entre nodos, es diferente a 1.80, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.746072**
SD: **0.2104131**
SE (Mean): **0.02104131**
CV: **0.1205065**

Calculamos intervalo de confianza:
Con $\mu \neq 1.80$
data: X5
t = -2.563, df = 99, p-value = 0.01188
alternative hypothesis: true mean is not equal to 1.8

95 percent confidence interval:
1.704322 1.787823

La distancia entre mejor unidad ganadora, es diferente a 1.80 de manera significativa dado que 0.01188 es < a 0.05. Ahora deseamos verificar si la distancia entre mejor unidad ganadora es mayor que 1.80.

Con $\mu > 1.80$
data: X5
t = -2.563, df = 99, p-value = 0.9941
alternative hypothesis: true mean is greater than 1.8
95 percent confidence interval:
1.711135 Inf

La distancia entre nodos es menor a 1.80, de forma no significativa (t = -2.563, df = 99, p-value = 0.9941) al 95% de confianza.

X6: Error de Cuantificación

Se verifica que el error de cuantificación generado por el algoritmo SOM-Kmeans es diferente a 0.03, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean:	**0.03237195**
SD:	**0.0009871452**
SE (Mean):	**9.871452e-05**
CV:	**0.03049385**

Calculamos intervalo de confianza:
Con $\mu \neq 0.03$
data: X6
t = 24.028, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.03

95 percent confidence interval:
0.03217608 0.03256782

El error de cuantificación, no es diferente a 0.03 de manera significativa dado que 2.2e-16 es < a 0.05. Ahora deseamos verificar el error de cuantificación es mayor que 0.03, tras ejecutar el algoritmo SOM-Kmeans.

Con $\mu > 0.03$
data: X6
t = 24.028, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is greater than 0.03
95 percent confidence interval:
0.03220804 Inf

El error de cuantificación, no es mayor que 0.03 de manera significativa a un 95% de confianza (2.2e-16 es < a 0.05).

Base plt

SOM-HC

X2: Tiempo de ejecución del algoritmo

Vamos a verificar que el tiempo de ejecución del algoritmo SOM-HC es diferente a 1.33, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.327018**
SD: **0.03724553**
SE (Mean): **0.003724553**
CV: **0.02806707**

Calculamos intervalo de confianza:

Con $\mu \neq 1.33$
data: X2
t = -0.80052, df = 99, p-value = 0.4253
alternative hypothesis: true mean is not equal to 1.33
95 percent confidence interval:
1.319628 1.334409

El tiempo de ejecución del algoritmo SOM-HC, no es diferente a 1.33 de manera significativa dado que 0.4253 no es < a 0.05. Ahora vamos a comprobar que el tiempo de ejecución del algoritmo sea mayor 1.33.

Con $\mu > 1.33$
data: X2
t = -0.80052, df = 99, p-value = 0.7873
alternative hypothesis: true mean is greater than 1.33
95 percent confidence interval:
1.320834 Inf

El tiempo de ejecución que da como resultado la ejecución del algoritmo SOM-HC, es menor o igual a 1.33 a un 95% de confianza, de manera no significativa (t = -0.80052, df = 99, p-value = 0.7873).

X4: Error topográfico - distancia entre nodos

Se desea verificar que la distancia entre nodos en la ejecución del algoritmo SOM-HC es diferente a 1.40, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.370854**
SD: **0.1290205**
SE (Mean): **0.01290205**
CV: **0.09411684**

Calculamos intervalo de confianza:

Con μ ≠ 1.40
data: X4
t = -2.259, df = 99, p-value = 0.02608
alternative hypothesis: true mean is not equal to 1.4
95 percent confidence interval:
1.345254 1.396455

La distancia entre nodos, es diferente de 1.40 de manera significativa dado que 0.02608 es < a 0.05. Ahora deseamos verificar si la distancia entre nodos es mayor a 1.40, tras ejecutar el algoritmo SOM-HC.

Con μ > 1.40
data: X4
t = -2.259, df = 99, p-value = 0.987
alternative hypothesis: true mean is greater than 1.4
95 percent confidence interval:
1.349432 Inf

La distancia entre nodos, es menor o igual a 1.40 a un 95% de confianza, de manera no significativa (t = -2.259, df = 99, p-value = 0.987).

X5: Error topográfico - distancia entre BMU

Se desea verificar que la distancia entre nodos, es diferente a 1.30, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.28358**
SD: **0.1115294**
SE (Mean): **0.01115294**
CV: **0.08688933**

Calculamos intervalo de confianza:
Con μ ≠ 1.30
data: X5

t = -1.4722, df = 99, p-value = 0.1441
alternative hypothesis: true mean is not equal to 1.3
95 percent confidence interval:
1.261451 1.305710

La distancia entre mejor unidad ganadora, no es diferente a 1.30 de manera no significativa dado que 0.1441 no es < a 0.05. Ahora deseamos verificar si la distancia entre mejor unidad ganadora es mayor que 1.30.

Con $\mu > 1.30$
data: X5
t = -1.4722, df = 99, p-value = 0.9279
alternative hypothesis: true mean is greater than 1.3
95 percent confidence interval:
1.265062 Inf

La distancia entre mejor unidad ganadora, no es mayor a 1.30 a un 95% de confianza, de manera no significativa (t = -1.4722, df = 99, p-value = 0.9279).

X6: Error de Cuantificación

Se verifica que el error de cuantificación generado por el algoritmo SOM-HC es diferente a 0.045, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **0.04262133**
SD: **0.009382374**
SE (Mean): **0.0009382374**
CV: **0.2201333**

Calculamos intervalo de confianza:
Con $\mu \neq 0.045$
data: X6

t = -2.5353, df = 99, p-value = 0.0128
alternative hypothesis: true mean is not equal to 0.045
95 percent confidence interval:
0.04075967 0.04448300

El error de cuantificación, no es diferente a 0.045 de manera significativa dado que 0.0128 es < a 0.05. Ahora deseamos verificar el error de cuantificación es mayor que 0.045, tras ejecutar el algoritmo SOM-HC.

Con $\mu > 0.045$
data: X6
t = -2.5353, df = 99, p-value = 0.9936
alternative hypothesis: true mean is greater than 0.045
95 percent confidence interval:
0.04106349 Inf

El error de cuantificación, es menor o igual a 0.045 a un 95% de confianza, de manera no significativa (t = -2.5353, df = 99, p-value = 0.9936).

SOM-Kmeans

X2: Tiempo de ejecución del algoritmo

Vamos a verificar que el tiempo de ejecución del algoritmo SOM-Kmeans es diferente a 1.30, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.314448**
SD: **0.02367532**
SE (Mean): **0.002367532**
CV: **0.01801161**

Calculamos intervalo de confianza:
Con $\mu \neq 1.30$

data: X2
t = 6.1025, df = 99, p-value = 2.04e-08
alternative hypothesis: true mean is not equal to 1.3
95 percent confidence interval:
1.309750 1.319146

El tiempo de ejecución del algoritmo SOM-Kmeans, es diferente de 1.30 de manera significativa dado que p-value es < a 0.05. Ahora vamos a comprobar que el tiempo de ejecución del algoritmo sea mayor 1.30.

Con $\mu > 1.32$
data: X2
t = -2.3451, df = 99, p-value = 0.9895
alternative hypothesis: true mean is greater than 1.32
95 percent confidence interval:
1.310517 Inf

El tiempo del algoritmo no es mayor a 1.32, de manera no significativa al 95% de confianza. El intervalo de confianza al 95% para μ es [1.309750, 1.319146] que no incluye al valor de 1.32. Según esto podemos decir que μ es menor.

X4: Error topográfico - distancia entre nodos

Se desea verificar que la distancia entre nodos en la ejecución del algoritmo SOM-Kmeans es diferente a 1.32, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.315966**
SD: **0.1162831**
SE (Mean): **0.01162831**
CV: **0.08836331**

Calculamos intervalo de confianza:

Con μ ≠ 1.32
data: X4
t = -0.3469, df = 99, p-value = 0.7294
alternative hypothesis: true mean is not equal to 1.32
95 percent confidence interval:
1.292893 1.339039

La distancia entre nodos, no es diferente a 1.32 de manera significativa dado que 0.5938 no es < a 0.05. Ahora deseamos verificar si la distancia entre nodos es mayor a 1.32, tras ejecutar el algoritmo SOM-Kmeans.

Con μ > 1.32
data: X4
t = -0.3469, df = 99, p-value = 0.6353
alternative hypothesis: true mean is greater than 1.32
95 percent confidence interval:
1.296659 Inf

La distancia entre nodos es menor o igual a 1.32, de forma no significativa (t = -0.346, df = 99, p-value = 0.6353) al 95% de confianza.

X5: Error topográfico - distancia entre BMU

Se desea verificar que la distancia entre nodos, es diferente a 1.30, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **1.256303**
SD: **0.09760472**
SE (Mean): **0.009760472**
CV: **0.07769205**

Calculamos intervalo de confianza:
Con μ ≠ 1.30

data: X5
t = -4.477, df = 99, p-value = 2.029e-05
alternative hypothesis: true mean is not equal to 1.3
95 percent confidence interval:
1.236936 1.275669

La distancia entre mejor unidad ganadora, es diferente a 1.30 de manera significativa dado que 2.029e-05 es < a 0.05.

Con $\mu > 1.30$
data: X5
t = -4.477, df = 99, p-value = 1
alternative hypothesis: true mean is greater than 1.3
95 percent confidence interval:
1.240096 Inf

La distancia entre mejor unidad ganadora, no es mayor a 1.30, de manera no significativa al 95% de confianza. El intervalo de confianza al 95% para μ es [1.236936, 1.275669] que no incluye al valor de 1.3. Según esto podemos decir que μ es menor.

X6: Error de Cuantificación

Se verifica que el error de cuantificación generado por el algoritmo SOM-Kmeans es diferente a 0.06, al 95% de confianza. Se calculan valores de media y desviación estándar:

Mean: **0.0645434**
SD: **0.004163309**
SE (Mean): **0.0004163309**
CV: **0.06450402**

Calculamos intervalo de confianza:

Con $\mu \neq 0.06$
data: X6
t = 10.913, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.06
95 percent confidence interval:
0.06371731 0.06536949

El error de cuantificación, no es diferente a 0.06 de manera significativa dado que 2.2e-16 es < a 0.05. Ahora deseamos verificar el error de cuantificación es mayor que 0.06, tras ejecutar el algoritmo SOM-Kmeans.

Con $\mu > 0.06$
data: X6
t = 10.913, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is greater than 0.06
95 percent confidence interval:
0.06385213 Inf

El error de cuantificación, no es mayor que 0.06 de manera significativa a un 95% de confianza (2.2e-16 es < a 0.05).

Identificación de patrones

Las variables utilizadas para la detección de patrones, serán unixtime y speed. Donde unixtime, corresponde a la hora en la que se sitúa un vehículo y speed la velocidad que tenía ese vehículo a esa hora.

Para la identificación de patrones, se definen las siguientes condiciones con respecto a la hora: si la variable unixtime, está dentro del rango de 0 a 5, se considera madrugada, de 6 a 12 se considera día, de 13 a 18 tarde y de 19 a 23 se considera de noche.

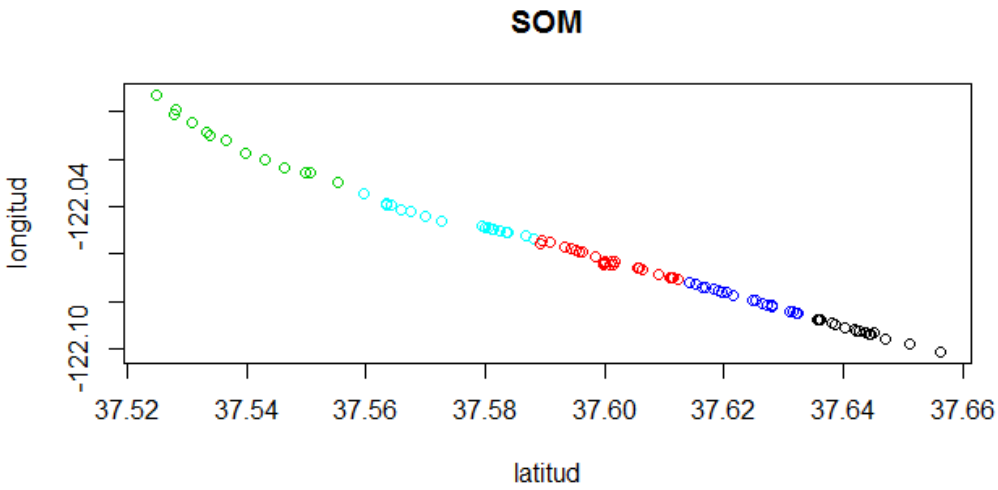
Combinación algoritmo SOM-K-means

Tiempo

Experimento #1

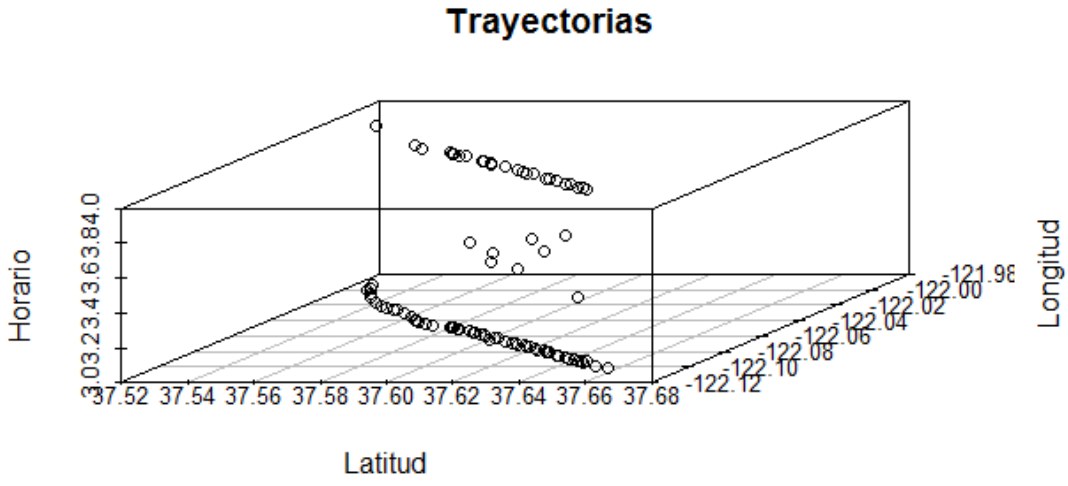
Experimento con topología 10x10, cinco cluster's, 100 épocas y con aprendizaje de 0.05.

GRÁFICO 19 : Experimento # 1 SOM-Kmeans plot 1



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

GRÁFICO 20 : Experimento # 1 SOM-Kmeans plot 2

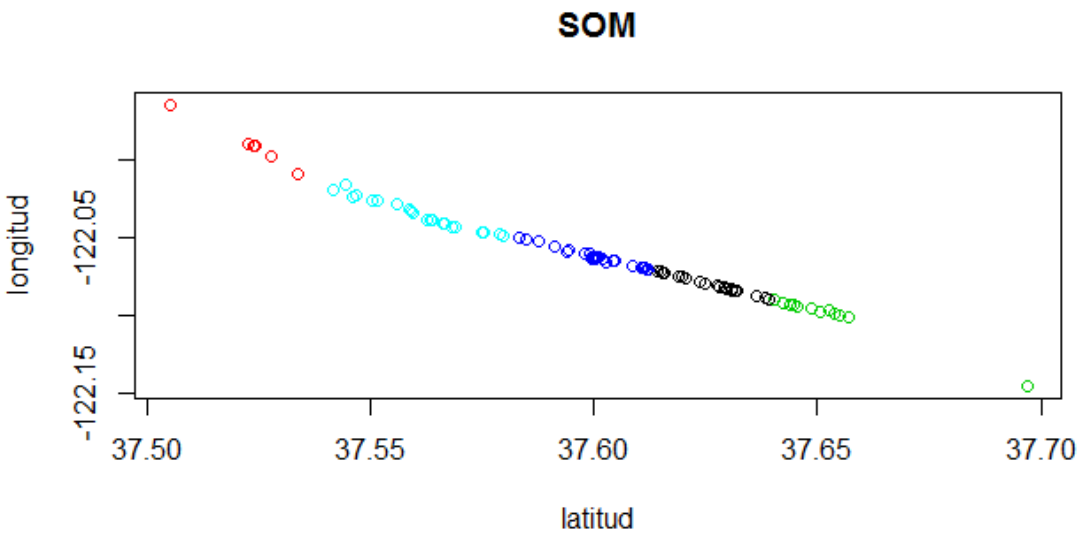


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #2

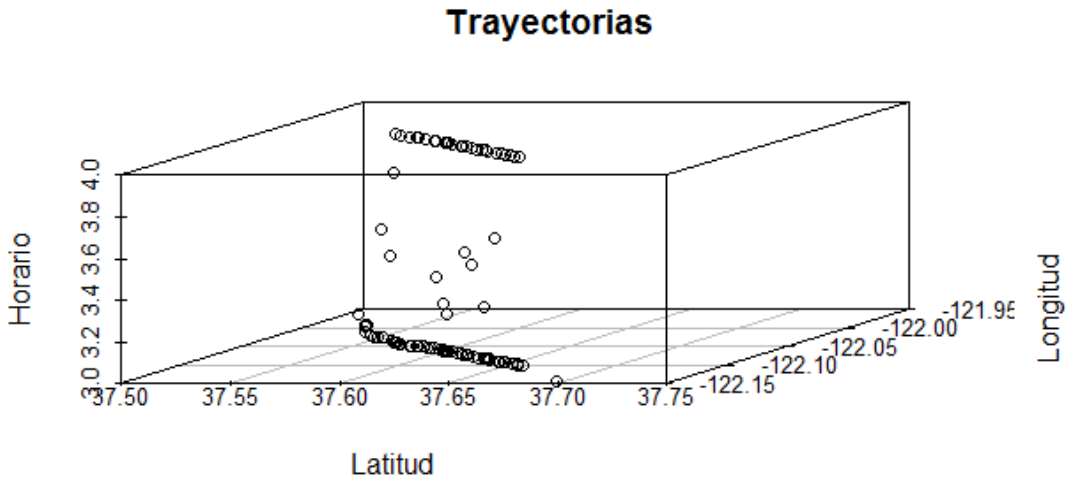
Experimento con topología 10x10, cinco cluster's, 200 épocas y con aprendizaje de 0.05.

GRÁFICO 21 : Experimento # 2 SOM-Kmeans plot 1



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

GRÁFICO 22 : Experimento # 2 SOM-Kmeans plot 2

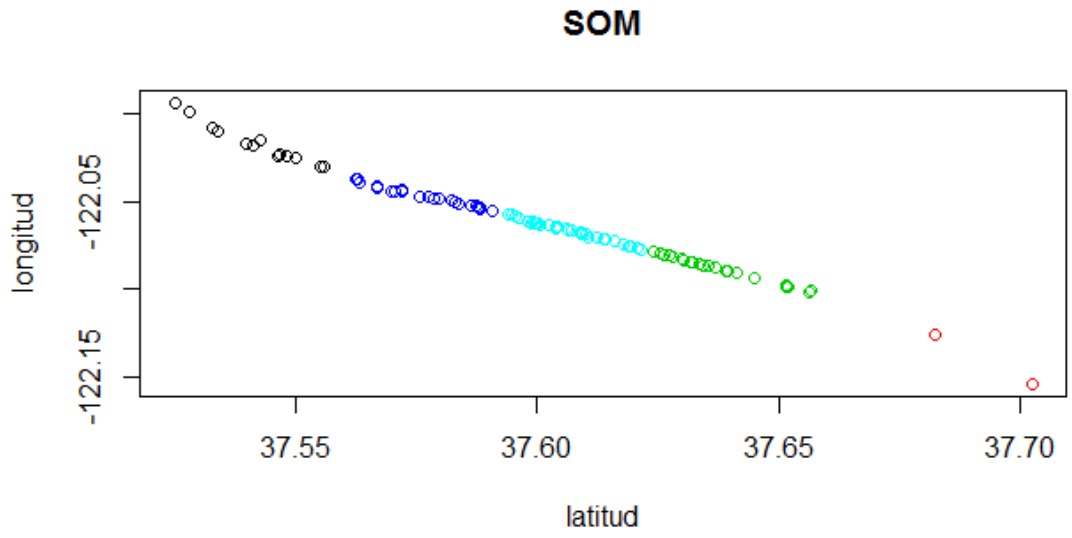


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #3

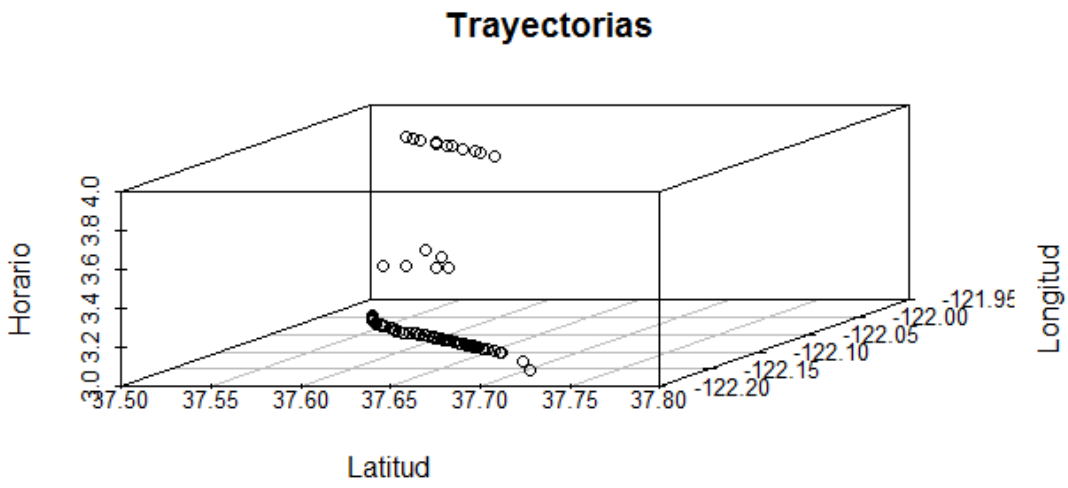
Experimento con topología 10x10, cinco cluster's, 500 épocas y con aprendizaje de 0.05.

GRÁFICO 23 : Experimento # 3 SOM-Kmeans plot 1



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

GRÁFICO 24 : Experimento # 3 SOM-Kmeans plot 2

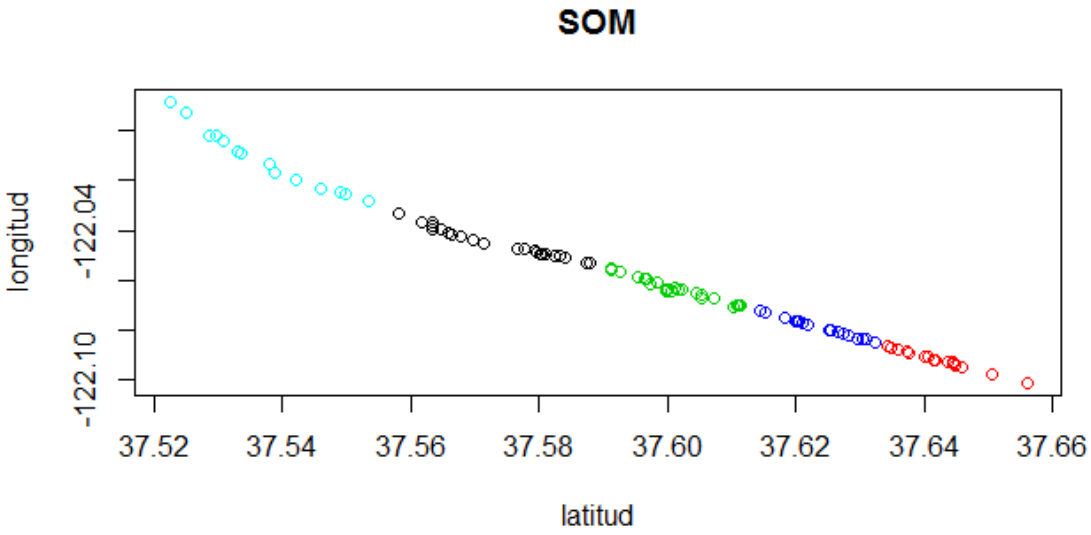


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #4

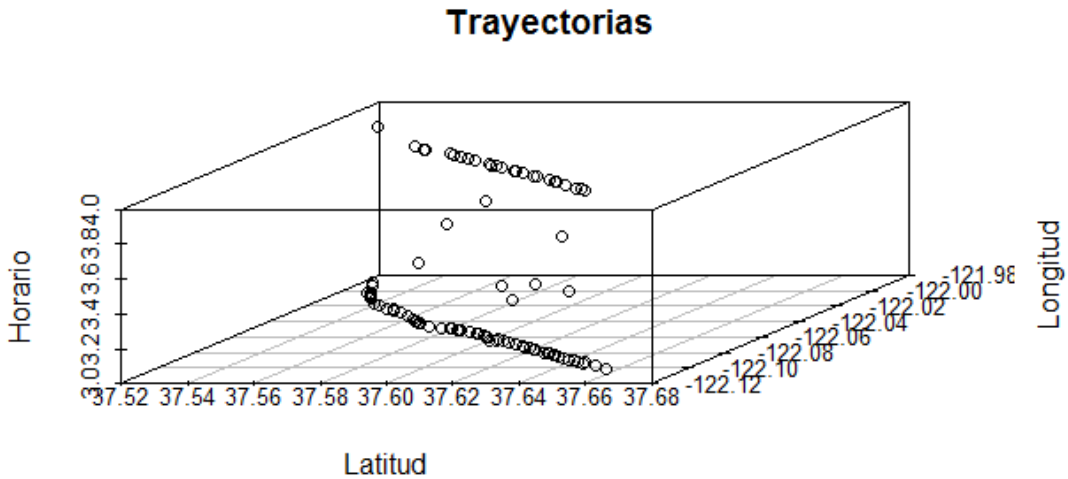
Experimento con topología 10x10, cinco cluster's, 100 épocas y con aprendizaje de 0.09.

GRÁFICO 25 : Experimento # 4 SOM-Kmeans plot 1



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

GRÁFICO 26 : Experimento # 4 SOM-Kmeans plot 1

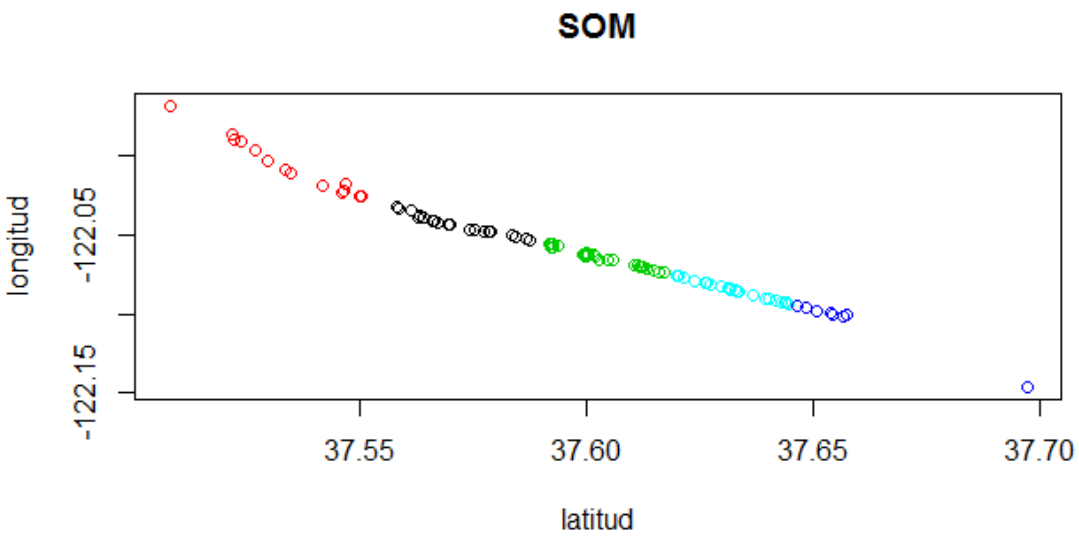


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #5

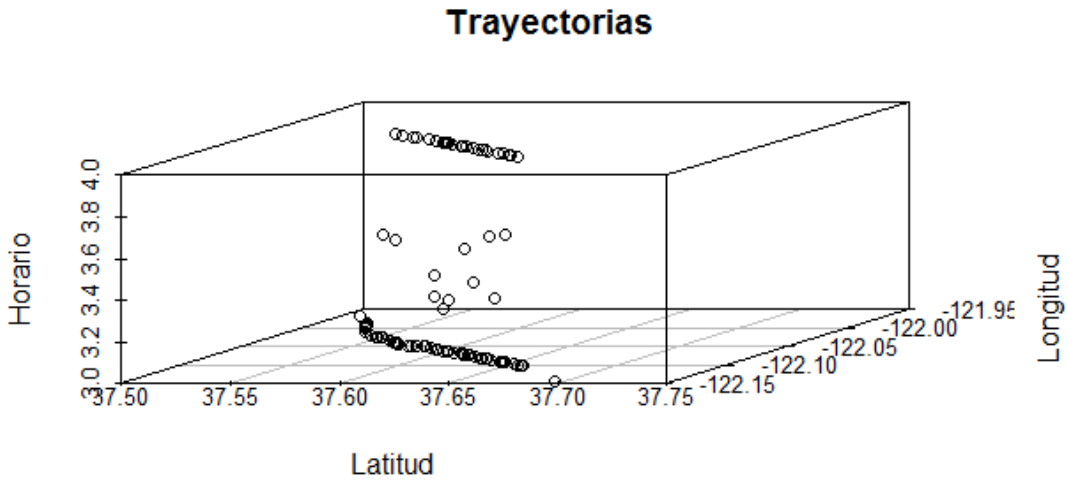
Experimento con topología 10x10, cinco cluster's, 200 épocas y con aprendizaje de 0.09.

GRÁFICO 27 : Experimento # 5 SOM-Kmeans plot 1



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

GRÁFICO 28 : Experimento # 5 SOM-Kmeans plot 2

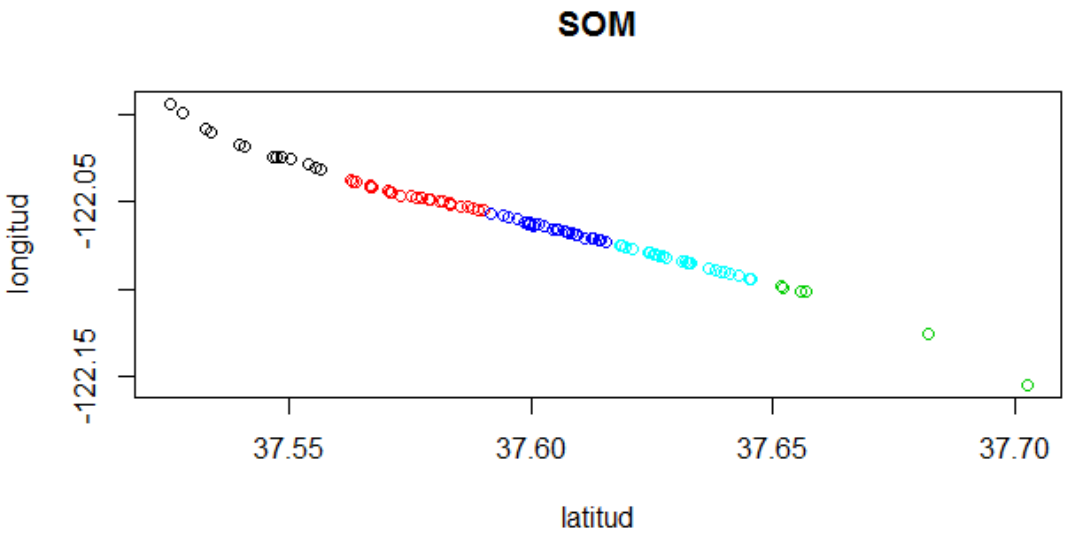


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #6

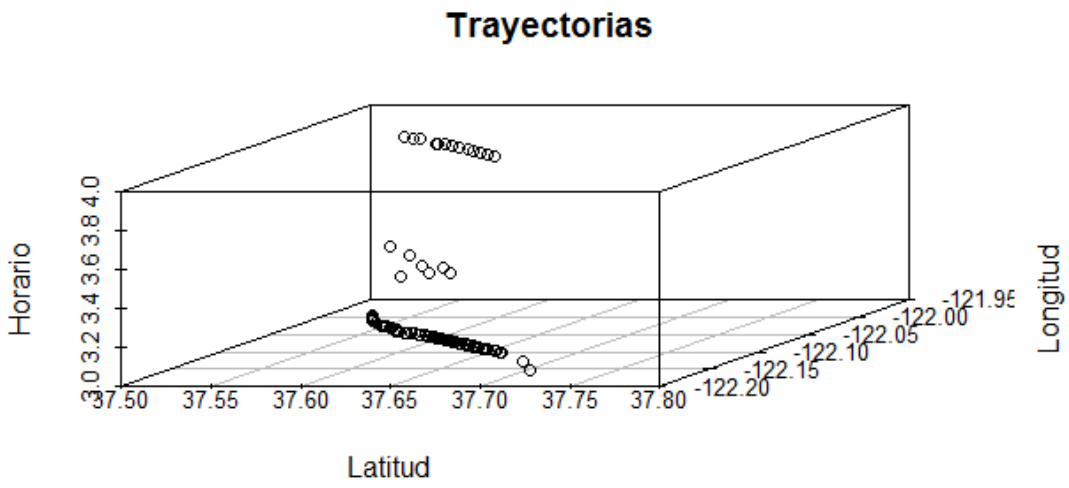
Experimento con topología 10x10, cinco cluster's, 500 épocas y con aprendizaje de 0.09.

GRÁFICO 29 : Experimento # 6 SOM-Kmeans plot 2



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

GRÁFICO 30 : Experimento # 6 SOM-Kmeans plot 2



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

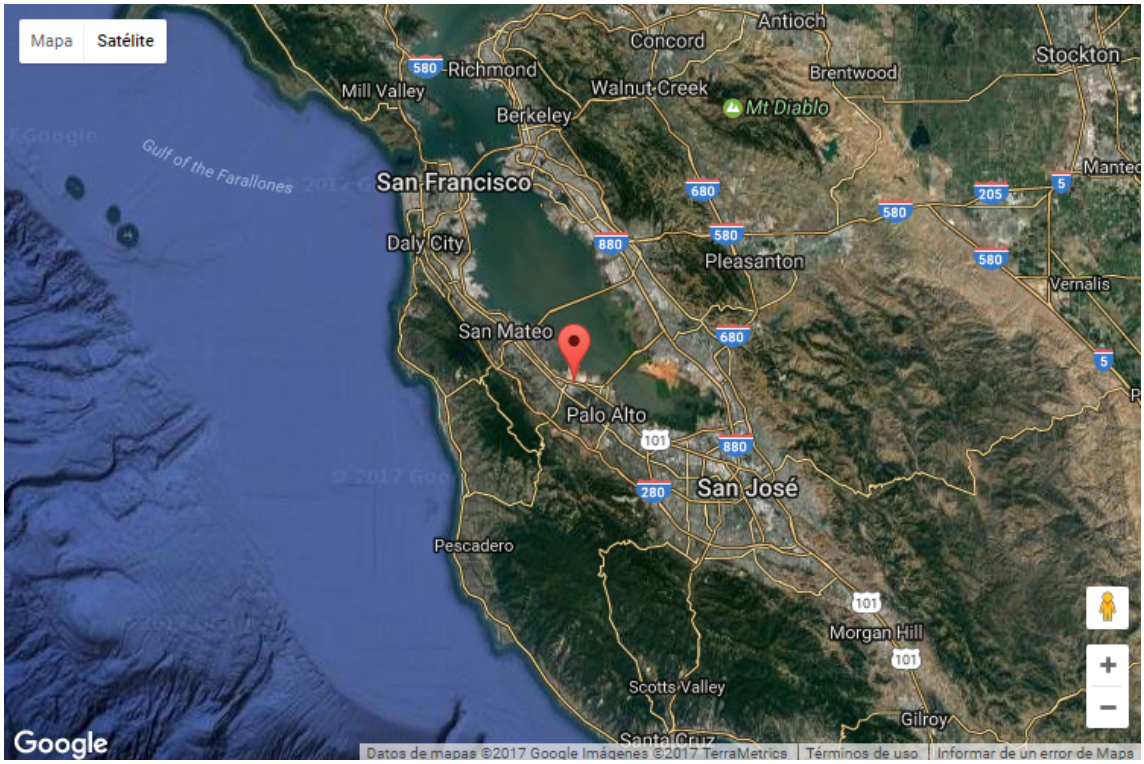
CUADRO 1 : Resultados: Identificación de patrones con SOM-Kmeans

Experimento	X1	X2	X3	X4	X5	X6
No. 1	8.271.028	2.515.404	0	1.285.471	1.288.939	0.03947535
No. 2	8.449.031	4.734.507	0	1.327.839	1.316.527	0.04312662
No. 3	8.992.031	210.037	0	1.475.802	1.303.452	0.04554874
No. 4	9.545.031	2.969.016	0	1.507.881	1.408.365	0.03908883
No. 5	9.634.032	5.336.008	0	1.353.425	1.417.226	0.04336917
No. 6	8.964.032	2.063.771	0	151.371	1.510.604	0.0439513

Elaborado por: Carlos Andrés Cervantes Suárez
Fuente: Resultados de la investigación

Según los experimentos realizados, con respecto a la identificación de patrones en base a al campo unixtime, se puede identificar lo siguiente:

GRÁFICO 31 : Mapa de california: Palo Alto

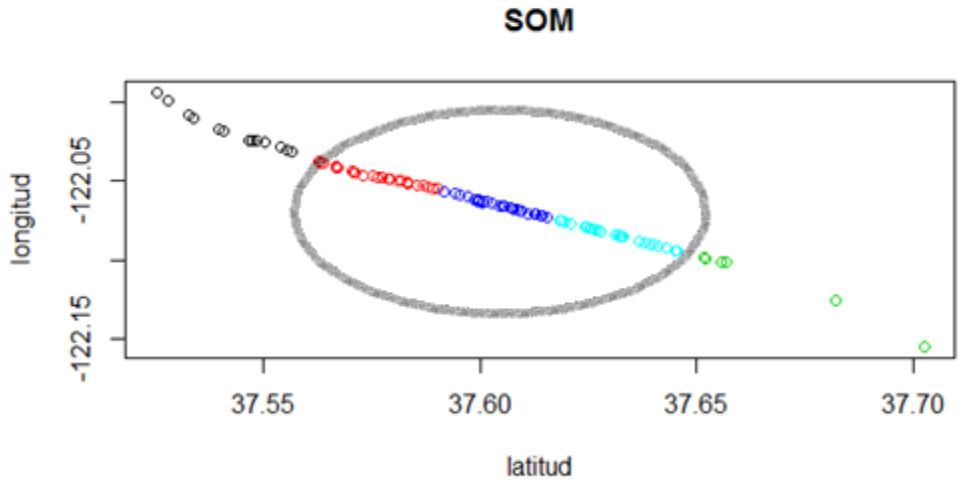


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Google Maps

Existen más puntos (vehículos en este caso), en los horarios tarde y noche; sin embargo la afluencia de vehículos es mayor en la tarde al Sur de San Francisco California.

Tomamos como referencia el experimento 6, ya que este para este experimento se consideró el mayor número de épocas para convergencia según la teoría revisada. Se puede visualizar que de los cinco grupos definidos, existen tres grupos con mayor afluencia de vehículos, en las siguientes coordenadas latitud 37.55 - 37.65 y longitud 122.05 - 122.10:

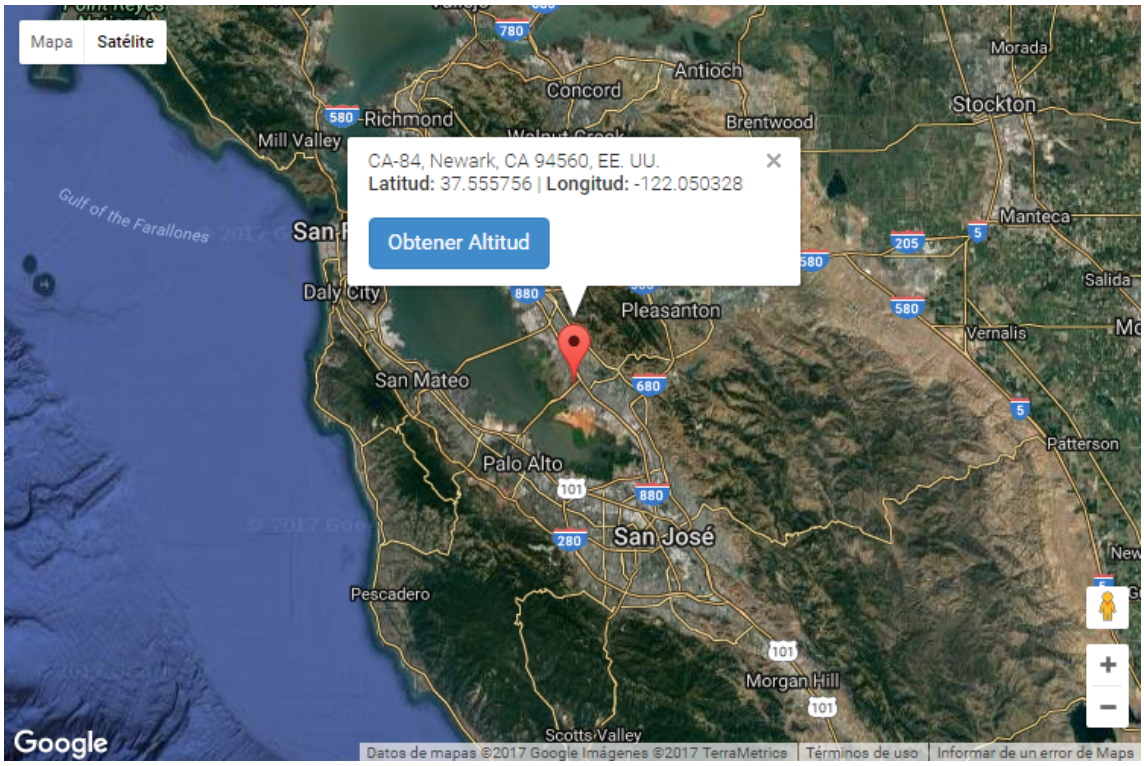
GRÁFICO 32 : Experimento SOM-Kmeans: Coordenadas con mayor afluencia de vehículos



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Visto desde el mapa de california, corresponde a la siguiente gráfica:

GRÁFICO 33 : Mapa de california: Coordenadas con mayor afluencia de vehículos



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Google Maps

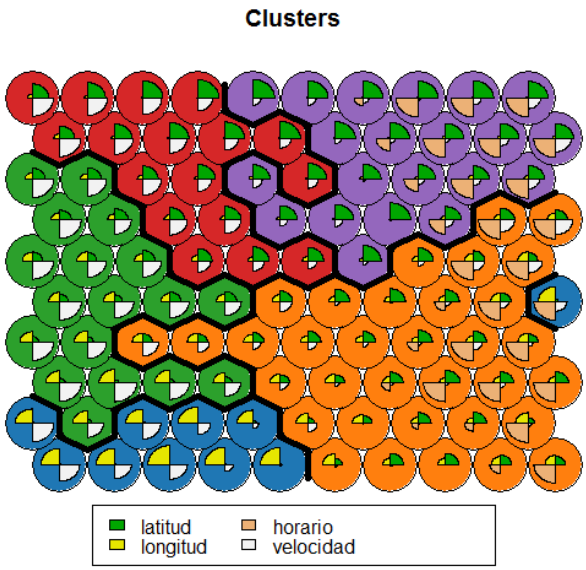
Combinación algoritmo SOM-HC

Con esta combinación se aprovecha se tiene la ventaja, de poder realizar un análisis usando todas las variables del modelo (latitud, longitud, tiempo y velocidad en este caso), de igual forma como se plantea en el trabajo de segmentación de clientes (Lynn, 2014).

Experimento #1

Experimento con topología 10x10, cinco cluster's, 100 épocas y con aprendizaje de 0.05.

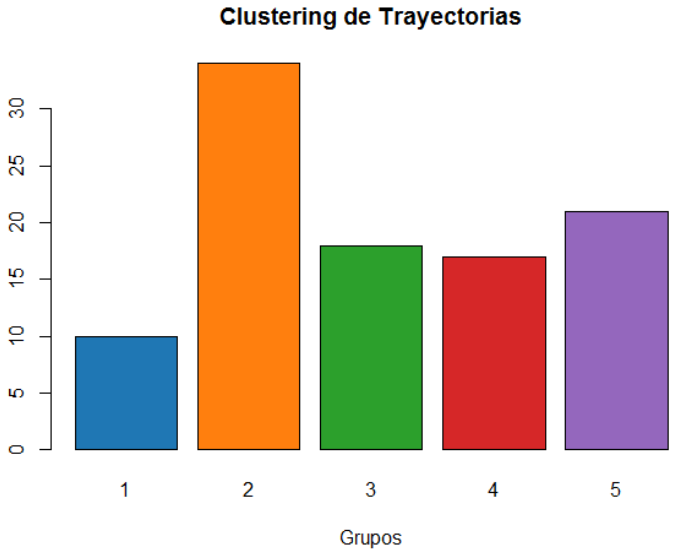
GRÁFICO 34 : Experimento # 1 SOM-HC plot 1



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Del mapa de SOM se puede identificar 5 grupos, de los cuales el más representativo es el número dos, los vehículos de este grupo tienen similitud en las características de ubicación (latitud y longitud). El grupo cinco tiene similitud con respecto a latitud en comparación con el primer grupo también se puede decir que ambos grupos comparten la característica de horario.

GRÁFICO 35 : Experimento # 1 SOM-HC plot 2

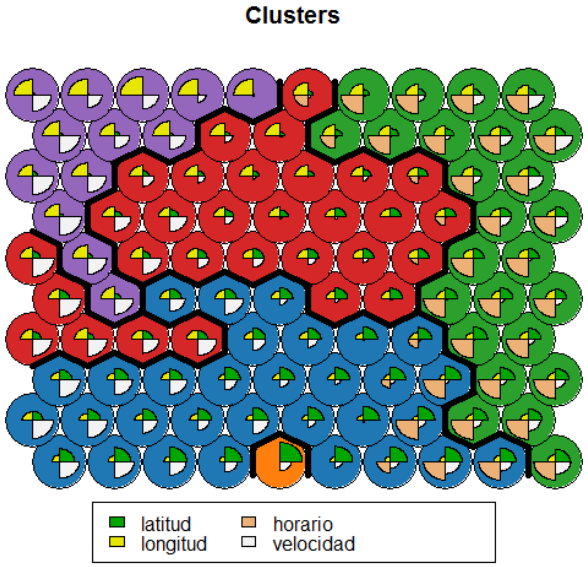


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #2

Experimento con topología 10x10, cinco cluster's, 200 épocas y con aprendizaje de 0.05.

GRÁFICO 36 : Experimento # 2 SOM-HC plot 1

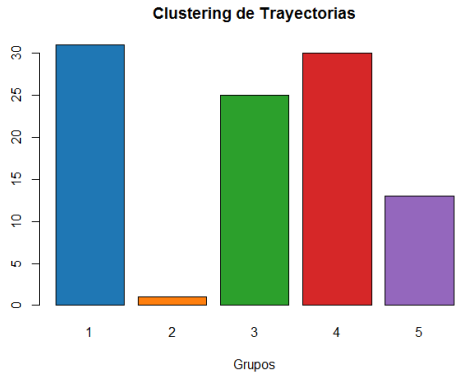


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

En este experimento, podemos visualizar que los grupos más representativos son el uno y el cuatro, de los cuales las características que comparten son la

latitud y la velocidad en el grupo uno. En el grupo cuatro las variables de similitud son latitud y longitud, eso quiere decir que los objetos de este grupo están ubicados en la misma zona geográfica.

GRÁFICO 37 : Experimento # 2 SOM-HC plot 2

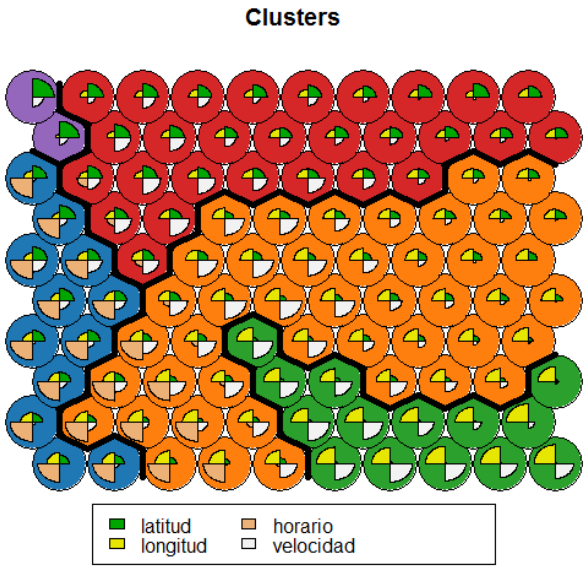


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #3

Experimento con topología 10x10, cinco cluster's, 500 épocas y con aprendizaje de 0.05.

GRÁFICO 38 : Experimento # 3 SOM-HC plot 1

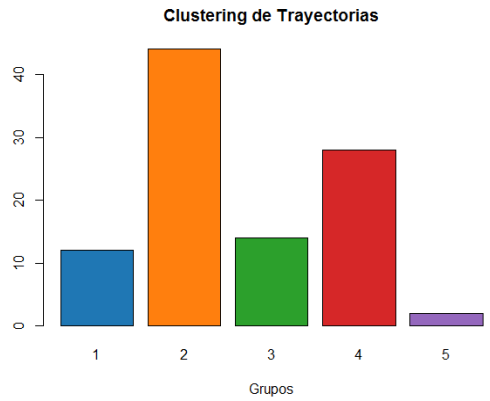


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Aquí se puede visualizar que de los 5 grupos definidos, los más representativos son el dos y el cuatro. Del dos se puede identificar que existen objetos

ubicados en la misma zona geográfica y otro grupo que comparten la misma velocidad y el horario.

GRÁFICO 39 : Experimento # 3 SOM-HC plot 2

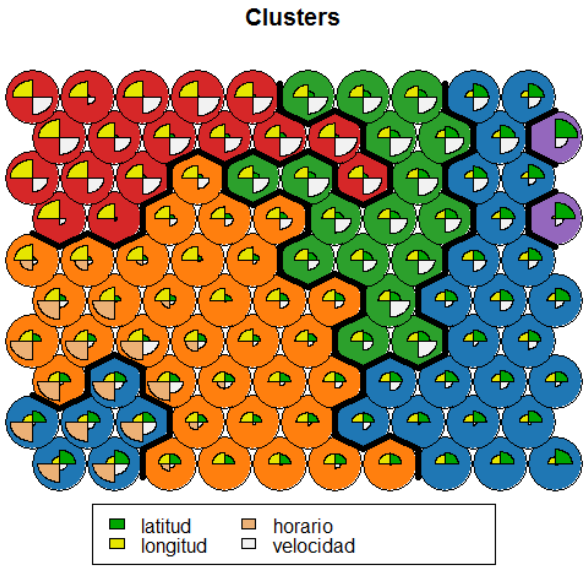


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #4

Experimento con topología 10x10, cinco cluster's, 100 épocas y con aprendizaje de 0.09.

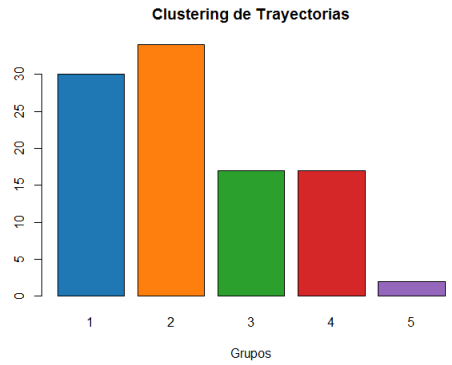
GRÁFICO 40 : Experimento # 4 SOM-HC plot 1



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

El grupo más representativo es el número dos por similitud de la característica de longitud, le sigue el grupo número uno por su similitud en latitud, longitud y velocidad.

GRÁFICO 41 : Experimento # 4 SOM-HC plot 2

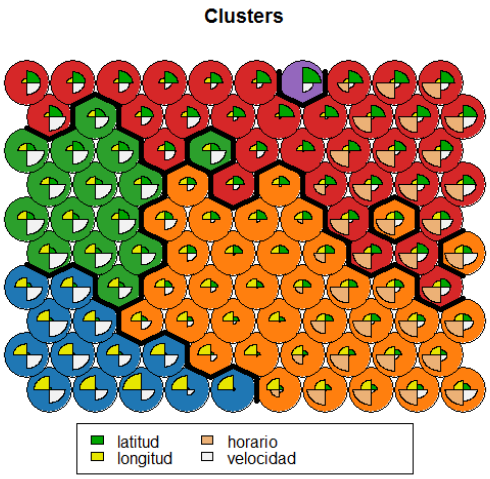


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #5

Experimento con topología 10x10, cinco cluster's, 200 épocas y con aprendizaje de 0.09.

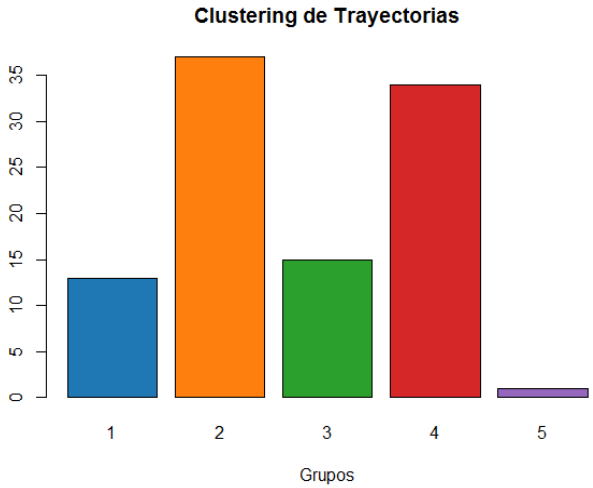
GRÁFICO 42 : Experimento # 5 SOM-HC plot 1



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Los grupos más representativos son el número dos por latitud y longitud es decir, estamos hablando de vehículos que se encuentran en la misma ubicación. La característica que sobresale del grupo número dos es la latitud.

GRÁFICO 43 : Experimento # 5 SOM-HC plot 2

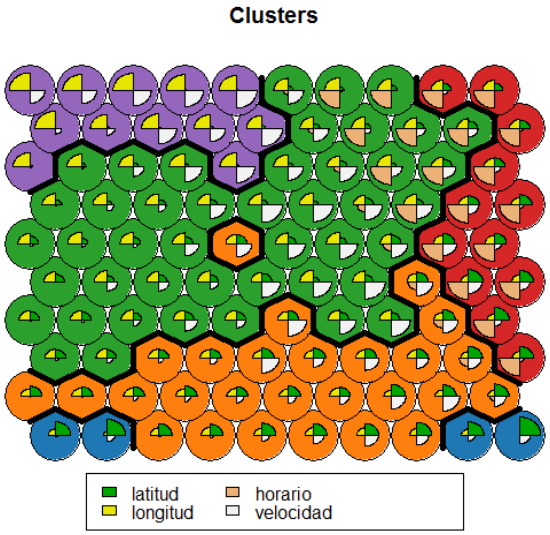


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

Experimento #6

Experimento con topología 10x10, cinco cluster's, 500 épocas y con aprendizaje de 0.09.

GRÁFICO 44 : Experimento # 6 SOM-HC plot 1

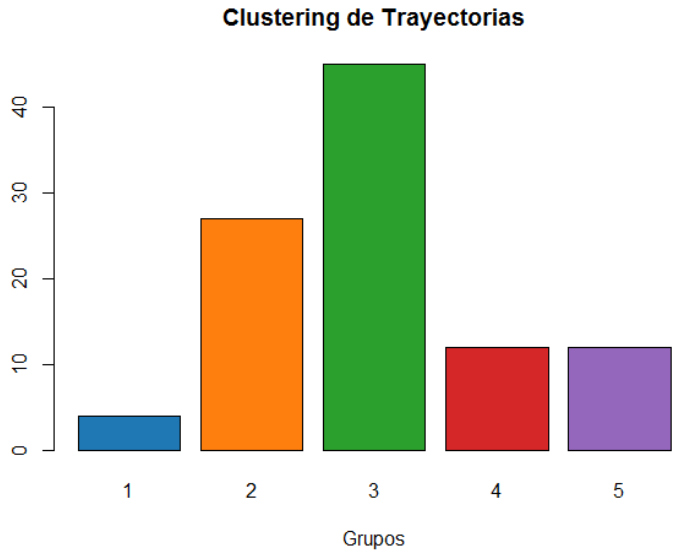


Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

En este experimento los grupos más representativos del mapa de características, son el tres y el dos. El tres contiene objetos que comparten la misma zona geográfica, en su mayoría, ya que también muestra, patrones de objetos que comparten la característica de velocidad y horario. El grupo dos

consta de un conjunto de vehículos que está ubicados en la misma zona geográfica.

GRÁFICO 45 : Experimento # 6 SOM-HC plot 2



Elaboración: Carlos Andrés Cervantes Suárez
Fuente: Experimento de la investigación

CUADRO 2 : Resultados: Identificación de patrones con SOM-HC

Experimento	X1	X2	X3	X4	X5	X6
No. 1	8.713.499	2.668.053	0.001000166	1.353.665	1.423.346	0.03950486
No. 2	8.805.504	4.809.775	0.0009999275	1.349.444	1.359.617	0.0426284
No. 3	8.268.473	1.920.176	0.001000166	1.373.912	1.343.246	0.04259826
No. 4	8.275.473	2.697.154	0.0009999275	1.497.345	1.528.055	0.04576071
No. 5	8.717.498	5.104.792	0	1.296.983	1.361.459	0.04230513
No. 6	8.281.474	1.916.693	0	1.590.134	1.506.341	0.04279175

Elaborado por: Carlos Andrés Cervantes Suárez
Fuente: Resultados de la investigación

Base California

Base california

De los experimentos realizados con la base california, el menor tiempo en ejecución del algoritmo corresponde a SOM – k-means, con un 10% de los datos tiene un tiempo de ejecución de 12 segundos. Le sigue SOM – HC con un tiempo 13 segundos de ejecución.

En esta ejecución de SOM – k-means, se obtuvo un error topográfico de 2.20 por distancia entre nodos y 2.17 por distancias entre mejor unidad ganadora. En SOM – HC, se obtuvo un error topográfico de 2.11 por distancia entre nodos y 2.04 por distancias entre mejor unidad ganadora. El error de cuantificación obtenido en ambas ejecuciones fue de 0.27. La topología de ambas ejecuciones fue de 6x7, con doscientas épocas para 5 clúster.

Con el 25% de los datos, el menor tiempo de ejecución lo tiene el algoritmo SOM – HC, con 19.08 segundos. El algoritmo SOM – k-means tiene una ejecución de 19.65 segundos. El tiempo de clustering en la ejecución con SOM – k-means, es de 0.01560903 segundos. El error de cuantificación en la ejecución de SOM – k-means fue de 0.2989925 y para SOM – HC fue de 0.3025025. La topología de ambas ejecuciones fue de 6x7, con cien épocas para 5 clúster.

Utilizando el 50% de los datos, el menor tiempo de ejecución es realizado por SOM – HC con 39 segundos, y 40 segundos el SOM – k-means. El error topográfico y de cuantificación en SOM – HC son menores con respecto a SOM – k-means. La topología de ambas ejecuciones fue de 6x7, con cien épocas para 5 clúster.

Con el 100% de los datos, SOM – k-means tiene un tiempo de ejecución de 1 minuto con 57 segundos y SOM – HC tiene un tiempo de 1 minuto con 37 segundos, El error topográfico en ambas ejecuciones es de 0.01298734. La topología de ambas ejecuciones fue de 6x7, con cien épocas para 5 clúster.

Cuando aumentamos la topología del mapa SOM el error disminuye, lo cual es ideal. Para las ejecuciones realizadas con una topología de 10x10, se obtuvo un error de cuantificación de 0.003. Con doscientas épocas, el error topográfico es menor con el SOM – k-means. El tiempo de ejecución del

algoritmo es de 6 minutos con SOM – k-means y de 8 minutos con SOM – HC. El tiempo de clustering en SOM – k-means es de un segundo, y en SOM – HC es cero.

Base t-drive

De los experimentos realizados con la base t-drive tenemos los siguientes resultados: Con el 10% de los datos, el algoritmo que tiene menor tiempo de ejecución es SOM – k-means con 6 segundos. El algoritmo SOM – HC, tiene un tiempo de ejecución de 7 segundos. El error de cuantificación es menor con SOM – HC: 0.06796404, a diferencia de SOM – k-means con 0.108308. El error topográfico es menor con SOM – k-means, tanto para la distancia entre nodos y la mejor unidad ganadora.

Con el 25% de los datos, el tiempo de ejecución de ambos algoritmos es de 19 segundos. El error de cuantificación con SOM – k-means es de 0.06251723, el cual es un valor mayor comparado con el error obtenido con SOM – HC, el cual es de 0.05606964. El error topográfico es mayor en SOM – k-means, en comparación con SOM – HC, con un mapa topológico de 6x7 y 100 épocas para 5 clúster.

Utilizando el 50% de los datos, SOM – k-means tiene un mejor tiempo de ejecucion: 38.48928 secs; sin embargo el error de cuantificación del algoritmo SOM – HC, es menor: 0.02531938, comprado con el de SOM – k-means: 0.03323267. En el error topográfico para el algoritmo SOM – k-means, se obtuvo que la distancia entre nodos es menor en comparación con SOM – HC, sin embargo la distancia entre mejor unidad ganadora, es menor en este último.

Con el 100% de los datos, obtenemos un rendimiento similar en ambos algoritmo que corresponde a un 1 minuto con 30 segundos. El error de cuantificación es de 0.02 en ambos algoritmos. El error topográfico, para distancias entre nodos es menor en SOM – k-means, con respecto al SOM – HC. Con la distancia de mejor unidad ganadora, el algoritmo SOM – HC, es menor.

Base plt

De los experimentos realizados con la base plt, con el 10% de los datos obtenemos un mejor rendimiento con el algoritmo SOM – k-means, ya que en tiempo de ejecución marca 6 segundos. El error de cuantificación, para ambos algoritmo marca 0.006. En el error topográfico tanto para distancia entre nodos y mejor unidad ganadora, el SOM – k-means tiene una mejor puntuación en comparación con el SOM – HC.

Con el 25% de los datos, se presenta un mejor rendimiento con el algoritmo SOM – HC. Se obtiene un tiempo de ejecución de 18 segundos, un error de cuantificación de 0.05. En el error topográfico, tenemos 1.17 y 1.06, para las distancias entre nodos y mejor unidad ganadora respectivamente.

Utilizando el 50% de los datos, en rendimiento obtenemos 38 segundos para SOM – k-means y 40 segundos para SOM – HC. En error de cuantificación, SOM – HC tiene un valor de 0.03894091 y 0.06089758 para SOM – k-means. En cuanto al error topográfico, se tiene menor distancia entre nodos con SOM – HC y menor distancia de mejor unidad ganadora con SOM – k-means.

Por último, con el 100% de los datos SOM – k-mean tiene mejor rendimiento, con tiempo de 1.280758 minutos, pero posee un error de 0.04248888, el cual es mayor comparado con el error obtenido con SOM – HC: 0.03907678. En los errores topográficos, se obtiene mejor rendimiento con SOM – HC.

CUADRO 3 : Resultados de la métrica: errores topográficos

Algoritmo	Base	Error Topográfico: nodedist	Error Topográfico: BMU
SOM-HC	california	1.21 - 1.23	1.17 - 1.21
SOM-HC	t-drive	1.667349 - 1.740028	1.508323 - 1.575807
SOM-HC	Plt	1.345254 - 1.396455	1.261451 - 1.305710
SOM-Kmeans	california	1.21 - 1.23	1.17 - 1.21

SOM-Kmeans	t-drive	1.733307 - 1.799837	1.704322 - 1.787823
SOM-Kmeans	Plt	1.292893 - 1.339039	1.236936 1.275669

Elaborado por: Carlos Andrés Cervantes Suárez
Fuente: Resultados de la investigación

CUADRO 4 : Resultados de performance de los algoritmos

Algoritmo	Base	Tiempo de ejecución	Error de Cuantificación
SOM-HC	california	1.28 -1.30	0.01
SOM-HC	t-drive	1.28 -1.30	0.02211094 - 0.02274046
SOM-HC	plt	1.28 -1.30	0.04075967 - 0.04448300
SOM-Kmeans	california	> 1.30	0.01
SOM-Kmeans	t-drive	> 1.30	0.03217608 - 0.03256782
SOM-Kmeans	plt	> 1.30	0.06371731 - 0.06536949

Elaborado por: Carlos Andrés Cervantes Suárez
Fuente: Resultados de la investigación

Es por estos resultados que nos permitimos concluir con que el algoritmo SOM no está orientado a la clasificación óptima de los datos, sino primordialmente para su monitoreo interactivo y similitud de características con los demás objetos del modelo. Se hace necesario un procesamiento o manipulación previa de los datos, antes de aplicar el algoritmo.

La creación de una red SOM es un proceso de aprendizaje no supervisado, que puede ser utilizado para detectar grupos en los datos de entrada además, identificar vectores de entrada que no están asociados al modelo.

A medida que aumentan la cantidad de datos, los valores de error de cuantificación y topografía, aumenta relativamente en el algoritmo SOM – k-means. El error topográfico decrece a medida que se utiliza más datos de trayectorias. El tiempo del algoritmo será mayor dependiendo de la dimensión del mapa SOM (la topología del mapa SOM).
El tiempo del algoritmo será mayor si aumentamos el valor de rlen (épocas), parámetro que indica cuantas veces será presentado el data set completo a la red. A medida que se realizan más iteraciones, el error de cuantificación relativamente decrece.

Es necesario recomendar que al conseguir una optimización del proceso de agrupamiento que admita una mayor escalabilidad del sistema, cuando la cantidad de trayectorias vehiculares crece. Para ello, es necesario la explotación del paralelismo computacional, esto haciendo uso de recursos con múltiples procesadores y mejores características. Es un punto clave para el estudio de estos algoritmos de agrupamiento en trabajos futuros.

Considerar en investigaciones futuras, el uso de otras distancias en vez de la euclidiana, para el cálculo de la mejor unidad ganadora en el clustering jerárquico. Esto con el propósito de identificar el comportamiento que se obtiene, si es beneficioso o no con respecto a las métricas de calidad de una red SOM.

Investigar sobre paquetes o librerías de mapas auto-organizados con crecimiento jerárquico (GHSOM), para lenguaje R, o en su defecto implementar el algoritmo. Evaluar el performance según las métricas de calidad de los mapas auto – organizados.

Bibliografía

BIBLIOGRAFÍA

A.Ultsch, H. S. (9-13 de Julio de 1990). Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. *Proceedings of the International Neural Network Conference (INNC-90)*.

Alahakoon, D. H. (1998). A structure adapting feature map for optimal cluster representation. *In International Conference on Neural Information Processing ICONIP98*, 809–812.

Andre Salvaro Furtado, R. F. (25-27 de Noviembre de 2012). M-attract: Assessing the attractiveness of places by using moving objects trajectories data. *XIII Brazilian Symposium on Geoinformatics*, 84–95. Campos do Jordão, São Paulo, Brazil.

Andrienko, N. V. (2011). Spatial generalization and aggregation of massive movement data. *IEEE, 17*(2), 205–219.

Arias, F. G. (2006). *El Proyecto de Investigación. Introducción a la metodología científica* (5 ed.). Caracas - Venezuela: Episteme.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford.

Bullinaria, J. A. (2004). Self Organizing Maps: Fundamentals. Introduction to Neuronal Networks: Lecture 16.

Chen, C. T. (1998). A Feedforward Neural Network with Function Shape Autotuning. *Neural Networks, 9*(4), 627-641.

Chih-Chieh Hung, W.-C. P.-C. (2015). Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. *The VLDB Journal, 24*(2), 169–192.

Cuadros-Vargas, E. (2004). *Recuperação de informação por similaridade utilizando técnicas inteligentes.* PhD thesis, Department of Computer Science - University of Sao Paulo. in portuguese.

da Silva, I. H. (2017). *Artificial Neural Networks A Practical Course.* Springer International Publishing.

Dayhoff, J. E. (1990). *Neural network architectures: An introduction.* New York: Van Nostrand Reinhold.

Demuth Howard, B. M. (1992). Neural Network Toolbox User's Guide.

F. Giannotti, M. N. (2007). "Trajectory Pattern Mining". *In Proceedings of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 330 – 339.

Fausett, L. (1994). *Fundamentals of Neural Networks: Arqchitectures, Algorithms and Applications.* Prentice-Hall.

Fayyad, U. M. (1996). Data Mining and Knowledge Discovery in Databases: Applications in Astronomy and Planetary Science. *In Proceedings of the thirteenth national conference on Artificial intelligence, 2*.

Forgy, L. E. (1965). Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics 21*, 768-769.

Fosca Giannotti, M. N. (2006). Efficient mining of temporally annotated sequences. *Proceedings of the Sixth SIAM International Conference on Data Mining*, 348–359.

G. Andrienko, N. A. (2007). "Visual Analytics Tools for Analysis of Movement Data". *ACM SIGKDD: 38-46, ISSN:1931-0145.*

G., S., M.V., G., & Carrillo H. (2002). ViBlioSOM: Visualización de Información Bibliométrica mediante el Mapeo Auto-Organizado. *Revista Española de Documentación Científica*, 477 - 484.

Galtung, J. (1971). *Teoría y Métodos de la Investigación Social.* (Eudeba, Ed.) Buenos Aires.

Hasperué, W. (2005). Mapas auto-organizativos dinámicos.

Hassoun, M. (1995). *Fundamentals of Artificial Neural Networks.* MIT Press.

Haykin, S. (1994). *Neural networks: a comprehensive foundation.* Prentice Hall.

Jain, A. K. (1996). Artificial neural networks: A tutorial. *IEEE Computer, 29*(3), 31–44.

Jain, A. K., Murty, M. N., & .Flynn, P. J. (1999). Data clustering: A Review. *ACM Computing Surveys, 31*(3), 264–323.

Jiménez-Andrade, J. L., Villaseñor-García, E. A., Escalera, N. M., Cruz-Ramírez, N., & Carrillo, H. (10-12 de Octubre de 2007). Una herramienta computacional para el análisis de mapas autoorganizados. *IEEE 5º Congreso Internacional en Innovación y Desarrollo Tecnológico*.

Kaski, S., Honkela, T., Lagus, K., & Kohonen, T. (6 de Noviembre de 1998). WEBSOM – Self-organizing maps of document collections. *Neurocomputing, 21*(1-3), 101–117.

Kaur, N., Sahiwal, J. K., & Kaur, N. (Mayo de 2012). Efficent K-means Clustring Algorithm Using Ranking Method In Data Mining. *ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology, 1*(3).

Kiviluoto, K. (1996). "Topology preservation in self-organising maps". *IEEE Int. Conf. on Neural Networks*, 294-299.

Kohonen, T. (1988). Self-organized formation of topologically correct feature. 509 - 521.

Kohonen, T. (1989). Self-organization and associative memory. *Springer*.

Kohonen, T. (1998). Self-organization of very large document collections: State of the art. (L. B. In Niklasson, Ed.) *Springer, 1*, 65-74.

Kohonen, T. (2001). Self-organizing maps. *Springer*.

Kung, S. (1993). *Digital Neural Networks.* Prentice-Hall.

Lagus, K. H. (1999). Websom for textual data mining. *Artificial Intelligence Rev, 13*(5-6), 345–364.

Langfelder, P., Zhang, B., & Horvath, S. (16 de Noviembre de 2007). Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut library for R. *Bioinformatics Advance Access*.

Lin, C. L. (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems.* Prentice-Hall.

Lynn, S. (20 de 1 de 2014). Self-Organising Maps for Customer Segmentation - Theory and worked examples using census and customer data sets. *Talk for Dublin R Users Group*.

Maren AJ, H. C. (1990). Handbook of Neural Computing Applications. *Academic Press*.
Martinetz, T. M., & Schulten, K. J. (1994). Topology Representing Networks. *Neural Networks, 7*, 507-522.
McQueen, J. (1967). Some methods for classication and analysis of multivariate observations. *Proceeding of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281-297.
Mesa, H., & Restrepo, G. (2008). On dendrograms and topologies. *MATCH Communications in Mathematical and in Computer Chemistry 60*, 371-384.
Michael Dittenbach, D. M. (24 - 27 de Julio de 2000). The Growing Hierarchical Self-Organizing Map. *Proceedings of the Int'l Joint Conference on Neural Networks (IJCNN'2000)*.
Pedreschi, F. G. (2008). "Mobility, Data Mining and Privacy: Geographic Knowledge Discovery". *Springer Verlag*.
Piatesky-Shapiro, G., & Frawley, W. (1991). Knowledge Discovery in Databases. *MA:AAA/MIT Press*.
Pölzlbauer, G. (2004). Survey and Comparison of Quality Measures for Self-Organizing Maps.
Postic, M., & Ketele, J.-M. d. (1992). *Observar las situaciones educativas.* Paris: Narcea.
R. Sotolongo, A., & Robles Aranda, Y. (Mayo-Agosto de 2013). INTEGRACIÓN DE LOS ALGORITMOS DE MINERÍA DE DATOS 1R, PRISM E ID3 A POSTGRESQL. *JISTEM: Journal of Information Systems and Technology Management*, 389-406.
Rich Elaine, K. K. (1994). *Inteligencia artificial.* McGraw-Hill.
S. Pressman, R. (2010). *Ingeniería del software: Un enfoque práctico* (7 ed.). McGraw-Hill.
Salvatore Orlando, R. O. (2007). Trajectory data warehouses: Design and implementation issues. *Journal of Computing Science and Engineering, 1*(2), 211–232.
Singh, K., Malik, D., & Sharma, N. (Abril de 2011). Evolving limitations in K-means algorithm in data mining and their removal. *IJCEM International Journal of Computational Engineering & Management, 2*.
Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning Series)*. MIT Press.
Swaminathan Sankararaman Pankaj K. Agarwal Thomas Molhave, J. P. (2013). Model-driven matching and segmentation of trajectories. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 234–243.
Tamayo, M. T. (2003). *El proceso de la Investigación científica* (4 ed.). (G. NORIEGA, Ed.)
Teuvo, K. (1990). The self-organizing map. *IEEE*, 1464 - 1480.
Trochin William, D. J. (2006). *The Research Methods Knowledge Base*. Atomic Dog Piblishing Inc.

Turban Efraim, R. S. (2011). *Desision Support and Business Intelligence Systems.* Financial Times Prentice Hall.

Turing, A. M. (14 de Agosto de 1952). The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 237*(641), 37-72.

U., F., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining.* Cambridge: Mass.: MIT Press/AAAI Press.

Venables, W. N., Smith, D. M., & R-Team. (2016). *An Introduction to R.*

Vesanto, J., & Alhoniemi, E. (Mayo de 2000). Clustering of the Self-Organizing Map. *IEEE Transactions On Neural Networks, 11*(3).

Carlos Andrés Cervantes Suarez, Ingeniero en Sistemas Computacionales (Universidad de Guayaquil) experiencia en área de desarrollo y arquitectura de software para Empresa de Telecomunicaciones. Conocimientos en metodologías y técnicas de diseño de sistemas distribuidos, implementación de aplicaciones java empresarial, experiencia en áreas de QA con herramientas de análisis de calidad del software como SonarQube. Conocimientos en Sistemas operativos Linux, herramientas open sources, administración y configuración de servidores de aplicaciones, base de datos relacionales y no relacionales. Cursos de CCNA (CISCO), actualmente realizando investigación en los campos de machine learning aplicados al análisis de trayectorias vehiculares.

Jeniffer Denisse Bonilla Bermeo, Magíster en Economía y Dirección de Empresas (Escuela Superior Politécnica del Litoral). Ingeniera en Ciencias Empresariales, concentración en Finanzas, mención en Dirección y Planeación comercial. Contadora Pública Autorizada; mención en Economía Empresarial, Finanzas Internacionales y Gestión Empresarial (Universidad de Especialidades Espíritu Santo). Jefe de Análisis de Riesgos en Institución Financiera, amplia experiencia en Crédito y Finanzas; capacitadora indoor a Nivel Nacional de Riesgos y Análisis Financiero, conocimientos en Estadística y Economía. Docente de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil. Cursos en metodologías ágiles y proyectos. Artículos científicos, Revista indexada a Scopus, 2017.

Christopher Gabriel Crespo León, Master en Sistemas de Información Gerencial. investigador en áreas de visión y robótica con publicación de artículos científicos, organización de eventos de difusión. Líder de arquitectura en Multinacional de Telecomunicaciones. Docente, investigador y coordinador del Área de software de la Facultad de Ciencias Matemáticas y Físicas en la Universidad de Guayaquil. Conocimientos en Sistemas Linux, Matlab, herramientas open sources, nodejs, servidores de aplicaciones. Cursos de CCNA (CISCO), SOA (ARCITURA), Administración e implementación de software para base de datos, SOA suite, weblogic y bus empresarial (Oracle), Administración de plataformas de facturación para TELCOS (Huawei y Tecnotree). Diplomado en gestión de Proyectos (Tecnológico de Monterey).

compAs

ISBN: 978-9942-770-35-6

compAs