Universidad de Alcalá

Escuela Politécnica Superior

GRADO EN INGENIERÍA INFORMÁTICA

Trabajo Fin de Grado

Sistema de seguimiento del patrón del comportamiento en consumo energético de clientes mediante técnicas de agrupamiento en línea

Autor: Juan Manuel Viera Marín

Tutor: David Fernández Barrero
Cotutor: José Lisandro Aguilar Castro

Universidad de Alcalá

Escuela Politécnica Superior

2022

ESCUELA POLITECNICA SUPERIOR

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

GRADO EN INGENIERÍA INFORMÁTICA

Trabajo Fin de Grado

Sistema de seguimiento del patrón del comportamiento en consumo energético de clientes mediante técnicas de agrupamiento en línea

Autor: Juan Manuel Viera Marin

Tutor: David Fernández Barrero

Cotutor: José Lisandro Aguilar Castro

TRIBUNAL:

Presidente: << Nombre y Apellidos >>

Vocal 1º: << Nombre y Apellidos >>

Vocal 2º: << Nombre y Apellidos >>

FECHA: << Fecha de depósito >>

Resumen

Este trabajo de fin de grado estudia distintos modelos de agrupamiento en línea aplicado a datos de consumo energético en edificios de carácter comercial. Estos modelos se caracterizan por ser capaces de realizar ejecuciones sobre una agrupación previa, en nuestro caso, mes a mes. Se estudian dos algoritmos de agrupamiento (LAMDA y X-means) capaces de variar y adaptar el número de clústeres en tiempo de ejecución. Se compara sus rendimientos y resultados finales, para finalizar analizando el comportamiento de los clústeres a lo largo de un año.

Palabras clave: Agrupación en línea, Aprendizaje automático, consumo energético, LAMDA, X-means

Abstract

This final degree project studies different online clustering models applied to energy consumption data in commercial buildings. These models focuses on performing clustering over previous clustering executions, in our case, monthly. Two clustering algorithms (LAMDA and X-means) capable of varying and adapting the number of clusters at runtime are studied. We compare their performance and final results, to finish by analysing the cluster behaviour over one year.

Key words: On-line clustering, Machine learning, energy consumption, LAMDA, X-means

Índice

Capítulo 1: Introducción

1.1 Contexto y motivación

En la actualidad, hay una inmensa demanda a nivel global de energía, la cual es necesaria para el consumo funcional de las tareas más generales de la vida, tales como la iluminación, el uso de equipos informáticos, electrodomésticos, y demás aparatos electrónicos. Los dispositivos anteriormente mencionados son vitales actualmente en nuestra sociedad.

Por otro lado, actualmente, los diferentes tipos de establecimientos (residenciales, comerciales, industriales, entre otros) están siendo dotados de dispositivos inteligentes, como cámaras, sensores, y distintos actuadores [1]. Estos dispositivos, en conjunto con la infraestructura de comunicación, caracterizan al paradigma Internet de las Cosas (IoT por sus siglas en inglés) [2]. Este paradigma se define como una red de objetos que toman datos de forma regular, los analizan, e inician una acción en función de un objetivo dado. El IoT se caracteriza por tener presencia en el entorno, conectando una amplia variedad de dispositivos entre sí de forma inalámbrica, los cuales interactúan y cooperan para proporcionar servicios [2].

Debido al aumento de energía requerida por este paradigma, el consumo en viviendas ha aumentado entre 1232 y 1460 kWh anuales [3]. Se estima que el consumo energético derivado de este aumento de dispositivos, en Europa pasará de los 4 TWh en 2015 a 104 TWh en 2025 [3]. Debido a este incremento en la demanda energética, existe una gran preocupación por alcanzar una mayor eficiencia y optimización del consumo [1], [4], [5]. Para ello, entre otras cosas, es necesario identificar el patrón de consumo de los usuarios, y a partir de esa información, plantear estrategias y mecanismos para ahorrar los recursos energéticos en la mayor medida posible.

1.2 Objetivos del proyecto

El objetivo de este trabajo es identificar el patrón de comportamiento de los clientes según sus perfiles de consumo energético. En particular, se necesita identificar como el patrón de comportamiento de los clientes va cambiando en el tiempo. Para ello, será necesario usar un sistema de aprendizaje en línea para reaccionar a los cambios en los datos en tiempo real.

Los objetivos específicos son:

1. Analizar el problema de identificación de patrones de comportamiento energético.
2. Desarrollar un sistema de aprendizaje automático en línea para determinar la evolución de los patrones de comportamiento energético de los clientes.
3. Realizar un conjunto de experimentos y analizar los resultados obtenidos.

1.3 Metodología

Para realizar este trabajo, se usará la metodología CRISP-DM (Cross Industry Standard Process for Data Mining), una metodología que divide el trabajo en fases, que será perfecta para el desarrollo del proyecto [6], [7]. Las fases del trabajo serán:

1. Comprensión del negocio: Entender los objetivos del proyecto desde la perspectiva del negocio (en nuestro caso, empresas energéticas y clientes) para caracterizar el problema. Las tareas específicas de esta fase son:
    a. Comprender el negocio
    b. Estudiar los requerimientos y recursos necesarios
    c. Determinar objetivos del proyecto
    d. Establecer un plan.
2. Comprensión de los datos: Requiere la recolección inicial de datos y la familiarización con los mismos. Aquí se deberán identificar potenciales problemas con los datos, detectar valores atípicos, analizar la distribución de los datos, entre otras cosas. Las tareas específicas de esta fase son:
    a. Recolectar los datos iniciales
    b. Describir las propiedades de los datos
    c. Explorar los datos
    d. Verificar la calidad de los datos
3. Preparación de los datos: Esta fase consta de todas las actividades necesarias para preparar el conjunto de datos final que será posteriormente analizado. Se deberán normalizar los datos y transformarlos, realizar un proceso de ingeniería de características para seleccionar, fusionar o generar más variables, entre otras cosas. Por último, en esta fase dividiremos los datos en 2, un conjunto se usará para realizar

el entrenamiento inicial del sistema, y un segundo conjunto para testear la llegada de datos en línea. Las tareas específicas de esta fase son:

   a. Seleccionar los datos que se usarán
   b. Preparar inicialmente los datos: normalización y limpieza de los datos, entre otras cosas.
   c. Realizar un proceso de ingeniería de característica para determinar las variables con las que se trabajará del conjunto de datos

4. Fase de modelado: En esta fase se implementan varias técnicas de modelado para el problema bajo estudio, En nuestro caso, se implementan varias técnicas de agrupamiento en línea para el problema de identificación de la evolución de patrones de consumo energético, usando el conjunto de datos previamente preparado. Las tareas específicas de esta fase son:
   a. Seleccionar técnicas de agrupamiento en línea
   b. Desarrollar los modelos de la evolución de patrones de consumo energético
5. Evaluación: Se usan los datos de prueba para simular la llegada de datos, y se evalúan los modelos previamente generados mediante el uso de métricas de rendimiento, para determinar la calidad de los modelos. Las tareas específicas de esta fase son:
   a. Calcular las métricas de calidad para los modelos
   b. Evaluar y comparar los modelos
6. Implantación: Esta fase tiene que ver con la implantación de los modelos en los sistemas de toda de decisión de la organización. Esta fase no será considerada en este proyecto, y particularmente, se sustituirá por la redacción de este manuscrito.

1.4 Aprendizaje Automático en el ámbito energético

Respecto al estudio del consumo de energía y sus características, la mayoría de los trabajos de investigación se centran en reducir el consumo y en optimizar el uso de los recursos energéticos. Por ejemplo, en [8] se centran especialmente en un uso eficiente de la energía y su infraestructura en ciudades inteligentes usando técnicas de aprendizaje automático. Los autores usan el aprendizaje automático para crear una red de aprendizaje profundo en una ciudad inteligente, para analizar y predecir el consumo energético de dispositivos con sensores IoT.

Otros artículos que analizan el consumo energético, estudian estrategias, realizan predicciones sobre el consumo energético, entre otras cosas [9], [10]. En [10] utilizan modelos de ML para predecir el consumo de un edificio inteligente, con el fin de la conservación de energía y la protección medioambiental. En ese estudio se desarrollan varios algoritmos para realizar la predicción de consumo (Regresión de Soporte Vectorial [11], Red neuronal artificial [12] y Bosques aleatorios [13]). En el estudio [9], se realiza una comparativa de distintos parámetros de configuración para modelos de eficiencia energética mediante el uso de ML. Plantean 2 escenarios, uno ignorando el desgaste de las plataformas energéticas y otro considerándolo. Para ambos casos, utilizan modelos de Regresión de Soporte Vectorial [14], Red neuronal artificial [15] y Regresión de Procesos Gaussianos [16]. En el artículo [17] se desarrolla un modelo predictivo basado en el consumo energético de los usuarios, que permite la monitorización y estimación del consumo energético del cliente final. En este caso, hacen uso del algoritmo K-means [18] y de Máquinas de Vectores de Soporte [19]. Por otro lado, para un análisis de las tendencias de los usuarios, actualmente existen diversos algoritmos y métodos, así como técnicas para reducir la complejidad del problema [20].

1.5 Estructura y contenidos

A partir del segundo capítulo de este trabajo, empezamos a desarrollar y aplicar las ideas expuestas en la introducción. Comenzamos contextualizando el paradigma de aprendizaje automático, dejando claras sus premisas, objetivos y principales características. Aquí veremos los tipos de aprendizaje automático existente, y explicaremos los algoritmos que hemos elegido para ejecutar nuestra agrupación en linea.

En el tercer capítulo desarrollaremos nuestro enfoque en profundidad, presentando nuestro conjunto de datos de prueba, y explicando cómo se realizarán las ejecuciones para un agrupamiento en línea. Posteriormente, realizaremos nuestra experimentación. Primero, expondremos las métricas seleccionadas para comprobar la validez y calidad de nuestros resultados, para lo cual hemos seleccionado métricas que validan la relación entre clúster y la solidez de cada clúster individualmente. En segundo lugar, se describirá cual ha sido el procedimiento de ejecución de ambos algoritmos, y las características específicas de cada ejecución. Este apartado termina con un análisis de los clústeres obtenidos.

Por último, el último capítulo presenta las conclusiones finales y futuros trabajos.

Capítulo 2: Marco Teórico

2.1 Introducción al Aprendizaje Automático

La cantidad de datos se ha disparado durante la última década, provenientes de diferentes fuentes, tales como las redes sociales, transacciones financieras, dispositivos, entre otros, lo que fundamenta la gran necesidad actual de asimilarlos y comprenderlos. Con esta nueva problemática, se han hecho necesarios mecanismos que permitan analizar y sintetizar los datos para poder interpretarlos, para identificar patrones, sacar conclusiones acertadas, entre otras cosas.

Por otro lado, el aprendizaje automático (*machine learning* en inglés) es un campo de la inteligencia artificial cuyo objetivo es desarrollar algoritmos que logren aprender [21]. Una de las ramas del aprendizaje automático son las estrategias que tratan de extraer conocimiento a partir de los datos. El uso de técnicas de aprendizaje automático se ha extendido a todos los sectores con distintos objetivos, desde clasificar gustos musicales, pasando por describir a los planetas, hasta la detección facial. Existen varios tipos de aprendizaje automático, pero los más conocidos son el supervisado y no supervisado.

En el *aprendizaje supervisado* se generalizan datos ya etiquetados (clasificados) en un modelo de reconocimiento/clasificación [21]. Así, estos algoritmos pasan por una fase de "entrenamiento" con los datos reales para construir el modelo de clasificación/reconocimiento, el cual después puede ser usado con otros datos reales en esas tareas de clasificación/reconocimiento. Es importante entender que en el aprendizaje supervisado todas las salidas posibles están controladas y determinadas [21]. Por ejemplo, si quiero clasificar plantas en base a sus características, debo proporcionarle al algoritmo todos los tipos de plantas que le estoy pidiendo que identifique para que él me clasifique los datos de forma correcta.

El *aprendizaje no supervisado* parte de la base de que los datos no están etiquetados y que va a realizar un proceso de análisis sobre ellos para identificar similitudes [21]. Este paradigma se centra en identificar similitudes en los datos para obtener conocimiento de esto, de utilidad cuando las categorías de los datos no están definidas. En el área del aprendizaje no supervisado, una de las técnicas que se usan son las de *agrupamiento* [21].

El objetivo principal del agrupamiento es separar los datos en subconjuntos más pequeños, llamados grupos (clústeres), tal que el contenido de los datos sea similar en cada clúster, pero diferente al contenido del resto de grupos. Según [22]-[25], un algoritmo de agrupamiento debe cumplir lo siguiente:

- Producir clústeres que, al añadir nuevos datos, tengan pocas posibilidades de ser alterados drásticamente.
- Pequeños cambios en las características de los datos no afecten significativamente al agrupamiento.
- La generación de clústeres es independiente del orden de los datos.
- Detectar y filtrar los valores atípicos o *outliers.*
- Ser escalable
- Capaz de definir clústeres con distintas formas
- Necesitar el conocimiento mínimo sobre los datos para determinar sus parámetros.

Existen varios tipos de algoritmos de agrupamiento, y cada uno de ellos puede generar clústeres distintos, incluso para el mismo conjunto de datos [22]. Algunos de ellos son:

- Algoritmos de agrupamiento particional: Estos algoritmos construyen particiones de datos, en los que cada partición representa un clúster. El

algoritmo más conocido y representativo de esta categoría es el K-means. Lo explicaremos más adelante, ya que lo usaremos en nuestro trabajo.

- Algoritmos de agrupamiento jerárquico: Son algoritmos que construyen una descomposición jerárquica del set de datos. En base a la jerarquía generada, el algoritmo puede ser clasificado como aglomerativo o divisivo. Se dice aglomerativo cuando cada dato se considera un clúster en sí mismo, y estos se unen sucesivamente hasta que se obtiene la estructura deseada. Al contrario, en los algoritmos divisivos, todos los objetos pertenecen a la misma estructura y se van dividiendo en sub-clústeres, hasta obtener la estructura deseada [26]. La mayoría de los algoritmos de agrupamiento aglomerativos son variantes del método de la distancia mínima o *single-link* [27], método de la distancia máxima o *complete-link* [28] y el método de la media o *average-link* [29].
- Algoritmos basados en densidad: Estos métodos consideran a los clústeres como regiones en las que la densidad de datos excede un valor predefinido [30], [31]. La idea de estos algoritmos es que los clústeres sigan creciendo siempre y cuando la densidad en la región supere el umbral definido. Un ejemplo de este tipo de algoritmo es el de agrupamiento espacial basado en densidad de aplicaciones con ruido o DBSCAN por sus siglas en inglés (*Density-Based Spatial CLustering of Applications with Noise*), propuesto por Ester, Kriegel, Sander y Xu en 1996 [32].
- Algoritmos basados en cuadrículas: En estos algoritmos se tiene la idea de dividir los datos entre un conjunto finito de cuadrículas, sobre el que se realizan las operaciones de agrupamiento. Su principal ventaja es su rápido

procesamiento [23]. Un ejemplo de esta clase es el de agrupamiento guiado genéticamente o *genetic-guided* [33].

- Algoritmos difusos o *fuzzy*: Estos algoritmos permiten que un dato pertenezca a uno o más clústeres, a diferencia del resto de algoritmos. También usaremos en este trabajo un algoritmo de esta categoría, el algoritmo de clasificación LAMDA [34].

2.2 Técnicas No supervisadas en línea

Debido a la naturaleza cambiante de los datos, usaremos técnicas de agrupamiento en línea, para adaptar el sistema a los cambios en los patrones de consumo de los usuarios, posibilitando actualizaciones en tiempo real. En este trabajo, usaremos una adaptación del algoritmo K-means y el algoritmo LAMDA.

K-Means

El algoritmo *k-means* es de los algoritmos más simples y comunes usados en agrupamiento, dividiendo el conjunto de datos en *k* clústeres. K-means intenta encontrar el centro de cada clúster, que es representante de una región de datos [21]. A este punto se le llama *centroide*. Así, K-means es una técnica de agrupamiento basada en centroides, este tipo de técnicas usan el *centroide* de un clúster para representar al mismo [23].

Este punto se puede calcular de varias formas, como la media o la mediana de los datos. Por otro lado, también se puede calcular la calidad del clúster con la varianza interna de este, esto es con la suma residual de cuadrados:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2 , \qquad \text{(Eq. 1)}$$

siendo p el punto en el espacio de un objeto, c_i el centroide del cluster C_i y la función *dist* (x, y) la distancia euclídea entre 2 puntos x e y [23].

K-means va alternando 2 pasos:

1. Asignación de puntos al centroide más cercano
2. Cálculo de centroides

Estos pasos se repiten en bucle hasta que los centroides se estabilizan y dejan de cambiar. En el siguiente ejemplo, sacado de la librería *mglearn* [21], veremos su funcionamiento en la Figura 1.

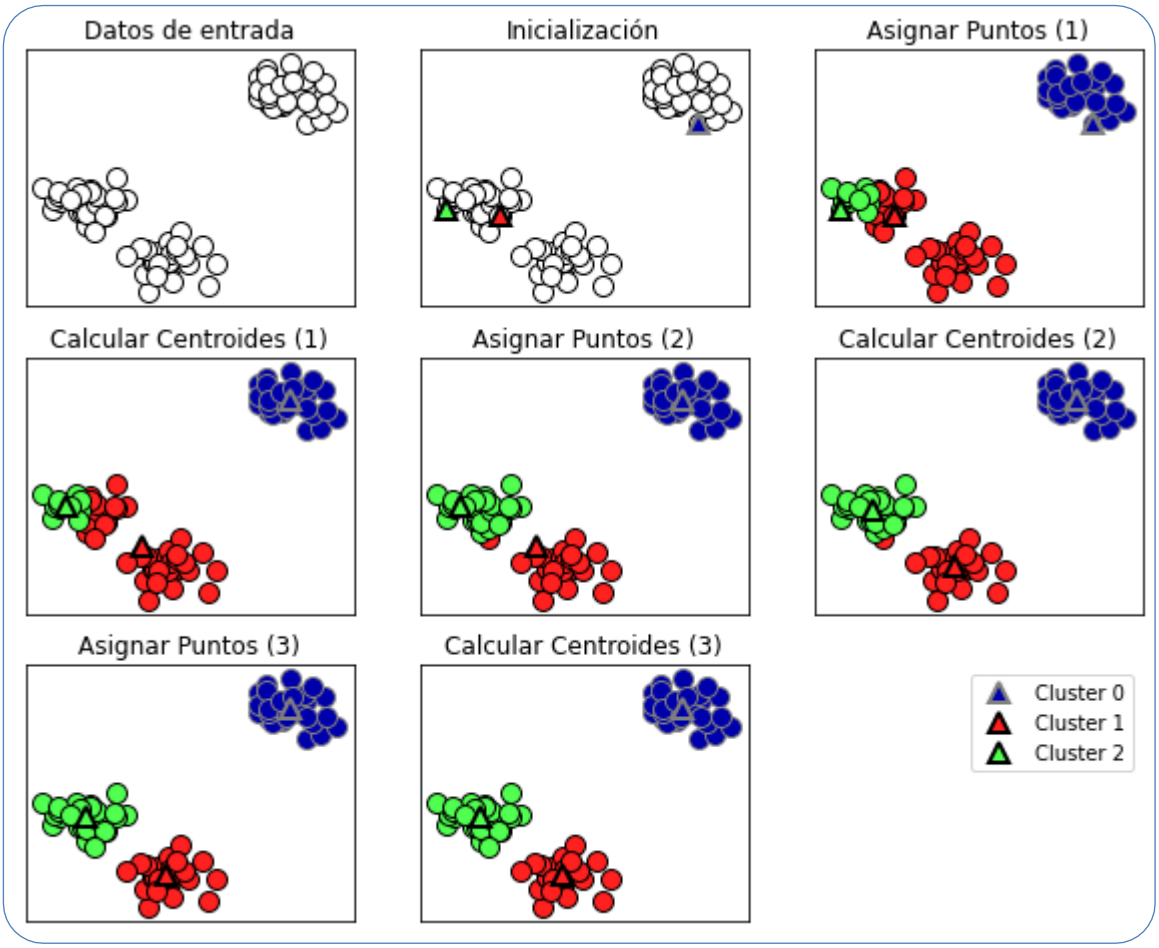

Figura 1. Ejemplo de ejecución paso a paso de K-means (tomado de [21]).

En este ejemplo se realiza una ejecución de k-means donde k = 3, dando lugar a 3 clústeres, los cuales estarán representados y diferenciados por un color. En este ejemplo vemos que se usan grupos bastante diferenciados entre sí, y con un centro relativamente definido. En el cuadro de **Inicialización**, vemos el conjunto de datos de entrada y los centroides iniciales seleccionados de forma arbitraria. Tras la primera asignación de puntos, vemos cómo se forman clústeres en función de su cercanía a cada centroide. Después de este primer paso, se vuelven a calcular los centroides, "centrándose" cada vez más en su grupo correspondiente. El proceso de asignación y cálculo de centroides se repite hasta el cuadro de **Calcular Centroides (3)** de la figura 1, donde los centroides no cambian respecto al cálculo anterior. En ese momento, el algoritmo converge.

El procedimiento del algoritmo se resume en la Figura 2 [23].

Entrada:

- *k:* número de clústeres.
- *D:* set de datos conteniendo *n* elementos

Salida: Set de *k* clústeres

Algoritmo:

(1) Seleccionar de forma arbitraria *k* elementos de *D*, a modo de centroides iniciales

(2) **repetir**

(3) Asignar cada elemento al centroide más próximo.

(4) Actualizar los centroides.

(5) **hasta que** no haya cambios en los centroides

Figura 2. Algoritmo de agrupamiento k-means

Particularmente, nosotros en este trabajo usaremos X-Means, que es una extensión de K-means que permite variar el valor de K. Así, X-means es un K-*means secuencial incremental* que determina el valor de K (clústeres) basada en una función *f(K)*, la cual es definida por la Ecuación siguiente [35]:

$$f(K) = \begin{cases} 1 & if K = 1 \\ \frac{S_K}{\alpha_K S_{K-1}} & if S_{k-1} \neq 0, \forall K > 1 \\ 1 & if S_{k-1} = 0, \forall K > 1 \end{cases} \quad \text{(Eq. 2)}$$

$$\text{Donde } \alpha_K = \begin{cases} 1 - \frac{3}{4N_d} & if K = 2 \wedge N_d > 1 \\ \alpha_{K-1} + \frac{1-\alpha_{K-1}}{6} & if K > 2 \wedge N_d > 1 \end{cases}$$

Donde S_k es la suma de las distorsiones del *clúster* cuando el número de grupos es K (ver más abajo), y N_d es el número de atributos del conjunto de datos (es decir, el número de dimensiones). El término $\alpha_k S_{k-1}$en la Ecuación anterior es una estimación de S_k basada en S_{k-1}, realizada bajo el supuesto de que los datos tienen una distribución uniforme. El valor de *f(K)* es la relación entre la distorsión real y la distorsión estimada, y es cercano a 1 cuando la distribución de datos es uniforme. Cuando hay áreas de concentración en la distribución de datos, S_k será menor que el valor estimado, de modo que *f(K)* disminuye. Cuanto más pequeño es *f(K)*, más concentrada es la distribución de datos. Por lo tanto, se puede considerar que los valores de K que producen un pequeño valor de *f(K)* proporcionan grupos bien definidos.

Por otro lado, La distorsión de un clúster es la distancia entre los objetos/individuos de un clúster y su centroide, según la siguiente Ecuación [35]:

$$I_j = \sum_{t=1}^{N_j} \left[d\left(x_{jt}, w_j\right) \right]^2 \quad \text{(Eq. 3)}$$

Donde I_j es la distorsión del cluster j, w_j es el centroide del cluster j, N_j es el número de objetos pertenecientes al cluster j, x_{jt} es el objeto perteneciente al clúster j, y $d\left(x_{jt}, w_j\right)$ es la distancia entre el objeto x_{jt} y el centroide w_j del cluster j. Cada clúster es representado por

su distorsión, y el impacto general de todos los clusters en todo el conjunto de datos se evalúa por la suma de todas las distorsiones, S_K, dada por la Ecuación siguiente [35]:

$$S_K = \sum_{j=1}^{K} I_j \quad \text{(Eq. 4)}$$

Donde K es el número de clústeres. Se supone que el número de clústeres K es mucho más pequeño que el número de objetos N. Particularmente, si para cualquier K inmediato *f(K)* muestra un comportamiento especial, en particular un punto mínimo, ese valor de K debería tomarse como el número deseado de clústeres. Así, X-Means converge cuando obtiene un valor mínimo de *f(K)*.

De esta manera, *X-means* determina si deben aparecer nuevos centroides dentro de un modelo actual (M_i). La aparición de nuevos centroides se lleva a cabo dividiendo algunos clusters en 2, los cuales han sido clasificados como optimizables según el criterio de *Schwarz* (es un criterio para la selección de modelos entre un conjunto finito de modelos), basado en el valor BIC, definido por la siguiente ecuación [36]:

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2}.logR \quad \text{(Eq. 5)}$$

Donde, $\hat{l}_j(D)$ es la probabilidad logarítmica de los datos en el modelo M_j; p_j es el número de parámetros libres presentes en el modelo M_j; y *R* representa el número de muestras presentes en *D* (*R* = |*D*|).

En esencia, X-means comienza con un *K* dado, y continúa agregando centroides (cambia el K) según el valor de *f(K)*, y calcula el score BIC para cada clúster para determinar, si fuese el caso, cual clúster dividir. Cuando X-Means converge (determina el valor ideal de K para ese conjunto de datos) se obtiene el agrupamiento final.

LAMDA (Learning Algorithm for Multivariate Data Analysis)

LAMDA es un algoritmo difuso no iterativo basado en el grado de adecuación de un dato a un grupo. Proporciona una gran versatilidad, ya que permite no especificar el número de clusters a la hora de ejecutar y, además, puede funcionar en línea [34]. LAMDA funciona realizando una evaluación de la similaridad entre los descriptores de un elemento X de la forma X = $\{x_1, x_2, \ldots, x_j, \ldots x_m\}$, que es su vector con m descriptores, con los descriptores de los centroides de los clusters existentes, para definir en que clúster se deberá introducir este dato X. Además, una vez que X se haya asignado a un clúster, este se convierte en X = $\{x_1, x_2, \ldots, x_j, \ldots x_m, c_i\}$, i = 1, 2, … k, donde c_j es la etiqueta asociada a X [37] .

A continuación, se resume las definiciones de base de LAMDA, obtenidas de [34], [37].

Normalización. Cada descriptor del elemento X debe estar *normalizado*, en base a sus valores máximos y mínimos:

$$\overline{x_j} = \frac{x_j - x_{jmin}}{x_{jmax} - x_{jmin}}, \qquad \text{(Eq. 6)}$$

donde $\overline{x_j}$ es el valor normalizado del descriptor j, x_{jmin} es el mínimo valor del descriptor j y x_{jmax} es el máximo descriptor del descriptor j. El elemento resultante de la normalización $\underline{X}$ será usado para computar el grado de adecuación del elemento a cada clúster existente.

Grado de Adecuación Marginal (MAD). Determina el grado de similaridad de un descriptor respecto a otro descriptor en una dada clase. Para el cálculo del MAD, se usan funciones de densidad, la más común, es la función binomial difusa:

$$MAD(\overline{x_j}/\rho_{kj}) = {\rho_{kj}}^{\overline{x_j}} (1 - \rho_{kj})^{(1 - \overline{x_j})}, \qquad \text{(Eq. 7)}$$

donde ρ_{kj} es el valor medio del descriptor j en el clúster k, calculado mediante:

$$\rho_{kj} = \frac{1}{n_{kj}} \sum_{t=1}^{n_{kj}} \overline{x_j}\ (t),$$

ρ_{kj} se va actualizando progresivamente cada vez que un nuevo elemento se añade al cluster.

La función para $MAD(\overline{x_j}/\rho_{kj})$ es la función de densidad de la distribución binomial, la cual puede ser interpretada como la probabilidad de que el descriptor normalizado analizado pertenezca a un clúster j, dada su media ρ_{kj}.

Grado de Adecuación Global (GAD). Determina el grado de adecuación de una muestra a cada clúster existente, se calcula mezclando el MAD con funciones de agregación. Estas funciones son interpolaciones entre la t-norma (T) y la t-conorma (S), como el operador Dombi [38]:

$$T(a,b) = \frac{1}{1+\sqrt[p]{(\frac{1-a}{a})^p+(\frac{1-b}{b})^p}} \quad \text{(Eq. 8)}$$

$$S(a,b) = 1-\frac{1}{1+\sqrt[p]{(\frac{1-a}{a})^p+(\frac{1-b}{b})^p}} \quad \text{(Eq. 9)}$$

En la mayoría de los casos, se suele tomar p = 1 para obtener una aproximación cercana a un comportamiento lineal de la t-norma y la t-conorma [38].

Existe también un parámetro de exigencia $0 < \alpha < 1$, usado para calibrar los datos de partición difusa [39]. Si α = 1 entonces GAD se calcula como la t-norma, obteniendo una clusterización más estricta. Si α = 0 entonces GAD se computa como una t-conorma, dando lugar a una agrupación más permisiva. Así, α produce una interpolación lineal entre la t-norma y la t-conorma para calcular el GAD [40].

$$GAD_{\bar{X},k}(MAD_{k,1},\ldots,MAD_{k,1}) = \alpha\, T(MAD_{k,1},\ldots,MAD_{k,1}) + (1-\alpha)\, S(MAD_{k,1},\ldots,MAD_{k,1})$$

Por otro lado, cuando un dato no pertenece a ninguna clase, se crea una clase no informativa (NIC), que será un clúster nuevo. El GAD de los datos que entran al NIC se computa considerando que $MAD_{NIC\ j} = 0.5$, independiente del valor de $\overline{x_j}$:

$$GAD_{\bar{X},NIC} = \alpha\, T(0.5,\ldots,0.5) + (1-\alpha)\, S(0.5,\ldots,0.5)$$

Ese elemento que entra al NIC se convierte en el primer elemento del nuevo clúster.

Finalmente, la asignación de elementos a un clúster se realiza calculando el GAD máximo de todas las clases. El índice (in) corresponde al número de la clase donde el elemento será asignado:

$$\text{in} = \max\left(GAD_{1\,\bar{X}}, GAD_{k\,\bar{X}}, \ldots, GAD_{m\,\bar{X}}, GAD_{NIC\,\bar{X}}\right)$$

El procedimiento del algoritmo se resume en la Figura 3 [37].

Entrada:

- X*:* elemento conteniendo *n* descriptores

Algoritmo:

(1) Normalizar descriptores

(2) Calcular el MAD para los descriptores

(3) Calcular el MAD del NIC considerando que $\rho_{NIC} = 0.5$

(4) Calcular el GAD de cada clase

(5) Asignar el elemento al clúster correspondiente

Figura 3. Algoritmo de agrupamiento LAMDA

Capítulo 3: Enfoque propuesto

3.1 Instanciación de las Técnicas No supervisadas en línea

En este apartado, explicaremos como realizamos la instanciación y ejecución de las dos técnicas presentadas en la sección anterior.

Preparación de los datos

La primera tarea es dividir el conjunto de datos en varios ficheros por periodos de tiempo. En nuestro caso, se dividieron por meses o trimestres. Los subconjuntos de datos creados fueron tomados desde datos reales del dataset número 3 de [41]. A partir de los datos originales, se han generado más datos usando la distribución de cada variable del dataset, con la finalidad de aumentar la cantidad de datos para nuestra ejecución.

Ese dataset corresponde a datos tomados de un edificio comercial en el 2018. El edificio tuvo un consumo máximo horario de 48 W/m^2, y el consumo anual fue de 183.2 kWh/m^2 [41]. Cada variable del dataset fue tomada cada media hora a lo largo de un año, desglosando el consumo total en kW de la siguiente forma:

- Total
- Luz
- Bomba de calor
- Unidades de tratamiento del aire
- Bombas de circulación - Calefacción y agua caliente
- Bombas de circulación - Enfriamiento
- Refrigeradores de aire
- Ascensores

Ejecución de las Técnicas

Cada técnica realiza una agrupación de cada fichero etiquetado previamente por mes/trimestre. A ese paso de agrupación por fichero lo llamaremos una iteración temporal (por mes/trimestre) de las técnicas. Cada iteración tomará como agrupamiento inicial los clústeres obtenidos (resultado) en la iteración anterior, exceptuando la primera iteración, que realiza un agrupamiento inicial. De esta forma, simulamos la evolución de los datos para analizar el comportamiento en línea de las técnicas. Este proceso itera hasta recorrer todos los ficheros disponibles.

En un caso real, la ejecución del proceso de agrupamiento seria exactamente igual, pero se realizaría cada cierto tiempo según la variabilidad de los datos. Por último, con los clústeres que se van obteniendo a través de las iteraciones, y los clústeres finales, se calculan las métricas para analizar los resultados.

3.2 Experimentación

Descripción de métricas a utilizar

Para medir la calidad de los resultados obtenidos, debemos usar un grupo de métricas. En nuestro caso, usaremos 2 métricas para evaluar la calidad de los clústeres, concretamente, el coeficiente de Silhouette [42] y el índice de Davies-Bouldin [43].

Silhouette

El coeficiente de Silhouette es una medida de la cohesión de los clústeres. Determina el grado de similaridad entre los objetos del mismo clúster y la densidad de estos [20]. Para conseguir esta medida, se calcula la media de las proximidades entre sus elementos. Esta métrica es, por tanto, efectiva en situaciones donde los clústeres tienen forma circular [42] o están agrupados en torno a un punto.

En la figura 4 vemos como se calculan las medidas a un punto *i* en el clúster A, para lo cual se calcula su distancia a los elementos en el mismo clúster A y en otros clusters [42].

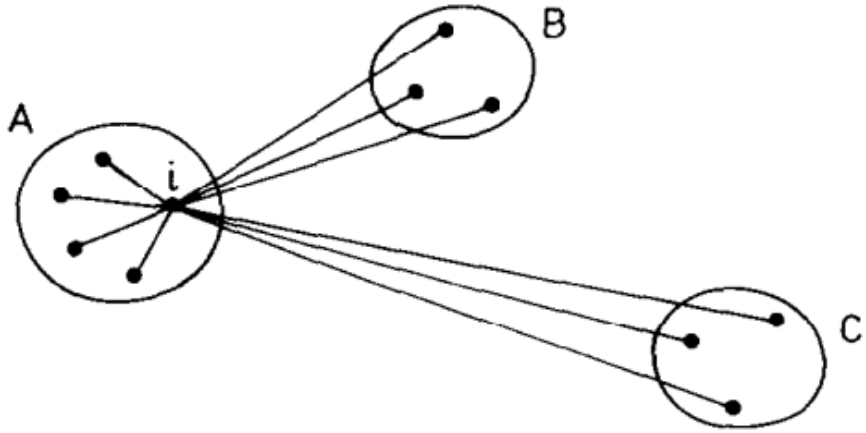

Figura 4. Ilustración de la computación de i en la construcción de siluetas (tomado de [42])

El coeficiente de silueta para una muestra de datos se determina con la media del coeficiente de silueta para cada dato de la muestra, calculado como [20]:

$$S_S = \sum_{i=1}^{n} \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \qquad \text{(Eq. 10)}$$

donde a(i) y b(i) son computadas para cada muestra i del clúster C_i ($i \in C_i$) para $a(i) = (|C_i| - 1)^{-1} \sum_{j \in C_i,\ i \neq j} d(i,j)$ y $b(i) = min_{k \neq i} |C_k|^{-1} \sum_{j \in C_k} d(i,j)$, donde d($i$, j) es la distancia entre los puntos *i* y *j* [20].

El coeficiente da un resultado entre -1 y 1. Los valores cercanos a 1 son los más óptimos, los cercanos a 0 indican que existen clústeres que se solapan, y los valores negativos generalmente indican que hay muestras asignadas a clústeres de manera errónea. Como norma general, cuanto mayor sea el coeficiente de silueta, mejor definido estarán los clústeres [20].

Davies-Bouldin

El índice de Davies-Bouldin se define como la similaridad media de cada clúster con su clúster más similar. Esa medida compara la distancia entre ambos clústeres con el tamaño de los clústeres en sí [20]. La medida se puede usar para deducir la adecuación de una partición de datos [43]. El índice de Davies-Bouldin se calcula como [20]:

$$S_{DB} = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j} R_{ij} \quad \text{(Eq. 11)}$$

Donde R_{ij} es la similaridad entre los clusters *i* y *j*. Hay distintas formas de calcular R_{ij}, una de ellas es $R_{ij} = {s_i + s_j}/{d_{ij}}$, siendo s_i la distancia media entre cada punto del clúster *i* y el centro de clúster *i* y d (i, j) la distancia entre los centroides de los clústeres *i* y *j* [20].

El valor mínimo que se puede obtener usando este índice es 0, que es el caso que se da cuando se forman tantos clústeres como individuos [50]. Por lo tanto, se entiende que los mejores valores de esta métrica son los más cercanos a 0, ya que indican una mejor partición y un modelo con una mejor separación entre clústeres [20].

Modelado

A continuación, se procede a describir como se obtuvieron los modelos y los resultados con cada técnica. El cálculo de las métricas para X-means se realizará usando la misma librería usada para su ejecución, en cambio, para las métricas de LAMDA se implementará el cálculo de ambas, ya que el formato del resultado de este algoritmo no nos permite usar librerías existentes.

X-means

Esta técnica fue implementada usando la *librería pyclustering* [44]. Siguiendo el procedimiento indicado en la sección anterior, se introducen secuencialmente los ficheros

etiquetados por mes/trimestes, según sea el caso, para simular esos periodos de tiempo. Cada uno de ellos representa una iteración temporal en la cual X-Means hace su respectivo agrupamiento. En la primera iteración *k* se inicializa en 3 (número de clústeres iniciales), valor que después X-Means optimiza en esa primera iteración (mes). En las siguientes iteraciones (meses), el algoritmo va reajustando ese valor de K según lo vea necesario. En el caso concreto del dataset usado, X-Means determina que son necesarios 20 clústeres en su primera iteración. Este número de clústeres se mantiene a lo largo de los 12 meses, X-Means determina que es el valor ideal de K en cada iteración (mes).

Empezaremos evaluando los centroides de los 20 clústeres de enero a diciembre en las Figuras 5 y 6. Observando ambas figuras, se puede ver que en un rango de aproximadamente 0.14 y 0.06 en los centroides (equivalente a un rango entre 200kW y 400kW sobre el total de kW), se encuentran 10 clústeres. También vemos 2 clústeres en el rango superior de la Figura 6, que destacan por estar separados de los de la zona media. Los clústeres 19 y 20 están aislados del resto a lo largo de toda la ejecución, convergiendo ligeramente y estabilizándose al final de esta. Particularmente, en la Figura 6 podemos ver que en los meses de verano los clústeres 19 y 20 se comportan de manera errática, quizá con un mayor número de clústeres este comportamiento se vería suavizado.

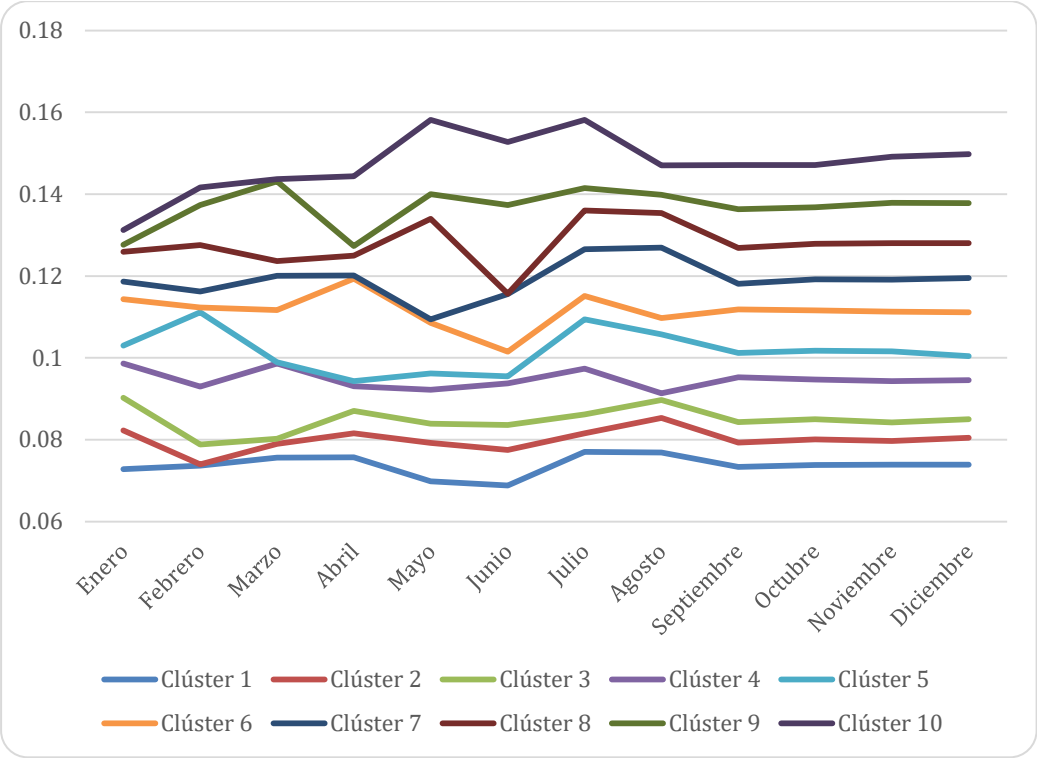

Figura 5. Evolución de los centroides de los primeros 10 grupos con X-means

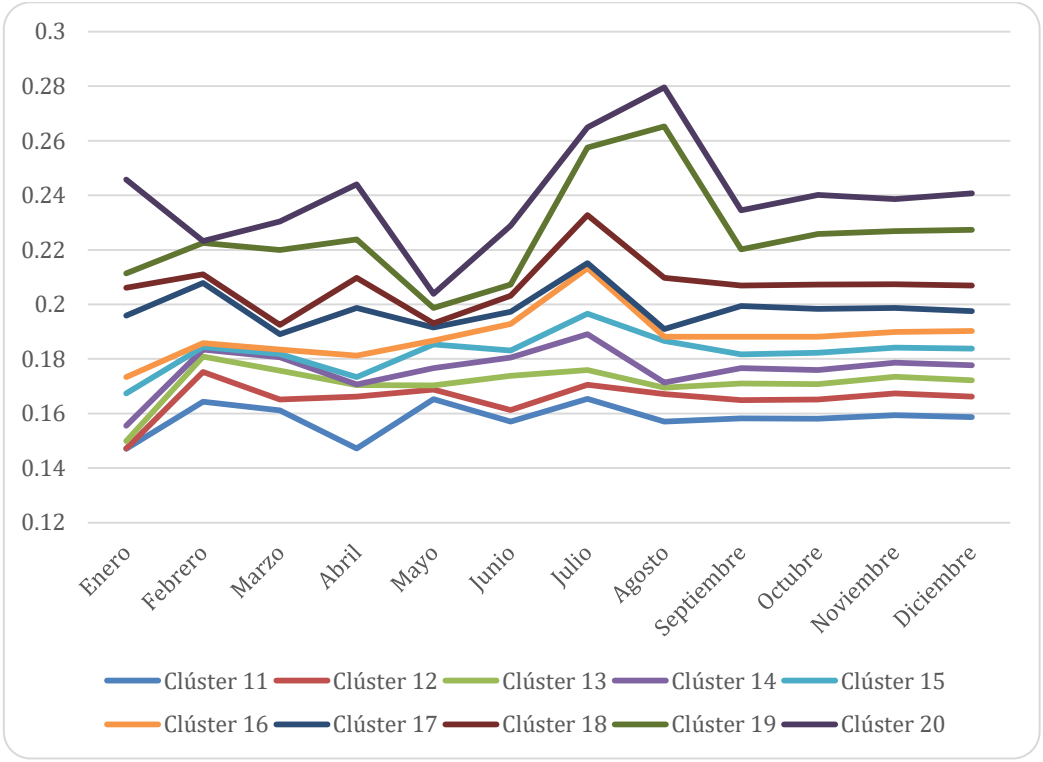

Figura 6. Evolución de los centroides de los últimos 10 con X-means

Acotando el rango superior de clústeres a 15, obtenemos una visión más detallada de los clústeres en la Figura 7. Nos encontramos 10 grupos que nunca superan 0.15 (500kW sobre el total), independientemente de la época del año. Por otro lado, podemos ver en la Figura 7 como en el último trimestre las variaciones son mínimas. Se puede deducir de esto que se ha llegado a un agrupamiento adecuado y estable, con unos grupos bien definidos.

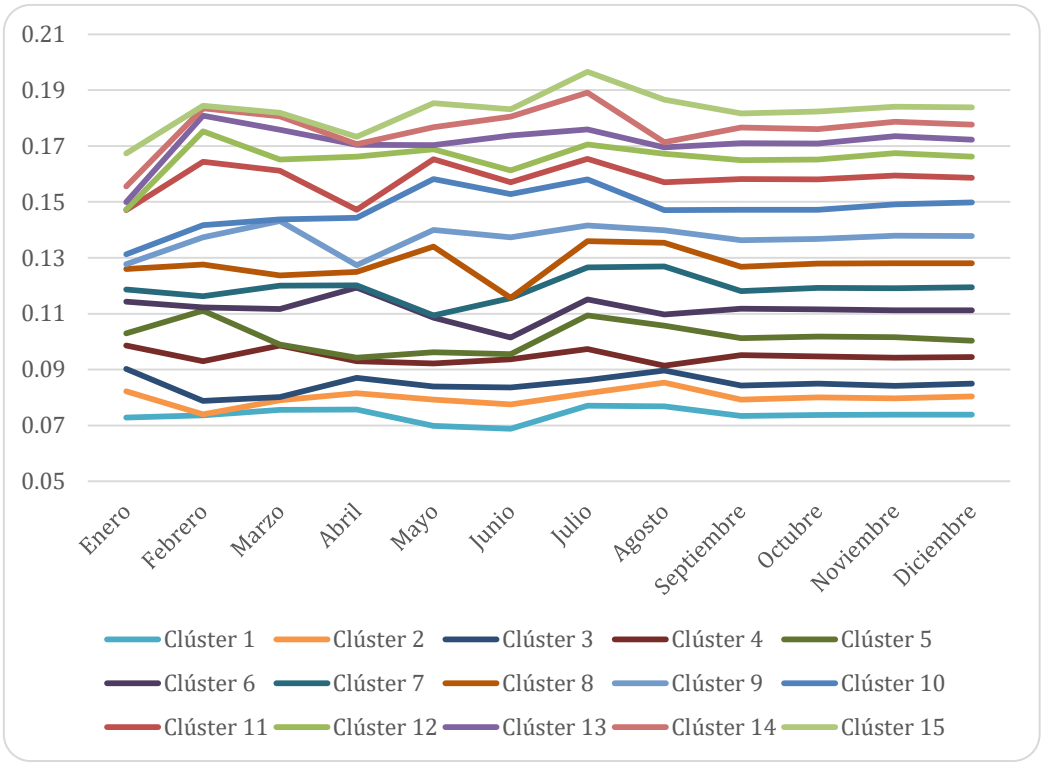

Figura 7. Agrupamiento acotado con X-means

LAMDA

Para la ejecución de LAMDA se ha usado una implementación de este algoritmo siguiendo lo indicado en los artículos [37], [44], y hemos realizado una adaptación del algoritmo a nuestro problema. Para realizar esta ejecución se han consolidado todos los datos

en un archivo añadiendo el descriptor de fecha y mes para poder evaluar los datos de forma correcta. De la misma forma que en la ejecución de X-means, se evalúan los datos mes a mes.

En la primera iteración de este algoritmo se inicia con un único clúster vacío, y se van creando nuevos clústeres cada vez que entra un elemento al NIC. Recordamos que los valores que entran al NIC son los que no han conseguido ser ubicados en clústeres existentes. Todos los valores son normalizados antes de iniciar su evaluación. LAMDA va eliminando, fusionando y creando clústeres dependiendo del GAD y el umbral de vecindad definido.

En la Figura 8 se puede ver como al inicio de la ejecución, en el mes de Enero, se crean 16 clústeres, aunque 3 de ellos (10, 11 y 13) se fusionan después del primer mes. Estos 13 clústeres restantes se mantienen a lo largo del resto de la ejecución. Todos los clústeres llegan a un punto distinto, a excepción de los conjuntos {7, 8} y el {4, 5} cuyos centroides terminan siendo bastante similares, aunque su trayectoria a lo largo de los meses es muy distinta.

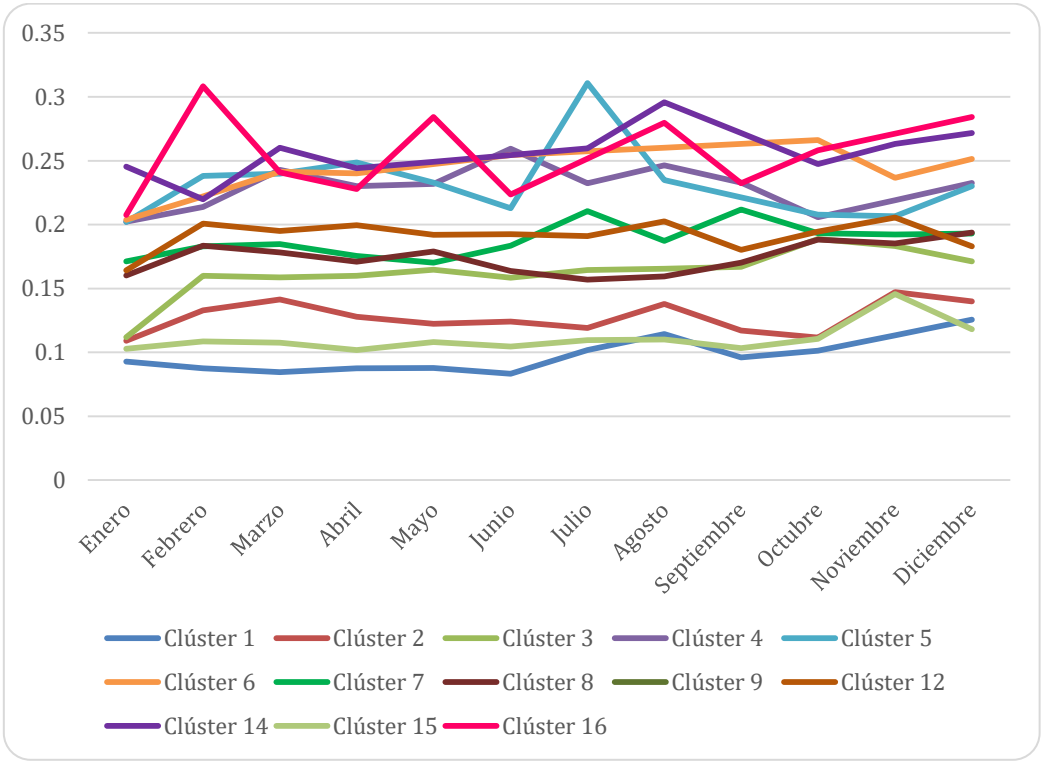

Figura 8. Evolución del centroide de los primeros grupos con LAMDA

En la Figura 9 vemos el resto de clústeres creados a lo largo de la ejecución. A partir de febrero se van creando nuevos grupos, y se puede ver que hay clústeres estables y otros que varían en el tiempo. Por ejemplo, el clúster 33 cambia completamente de tendencia, pasando de estar en un rango de 0.2-0.25 en julio a bajar a 0.09 en octubre, estableciéndose como el único clúster en ese bajo valor. Aquí también podemos ver el último clúster que se crea es en Agosto, siendo este el número 40.

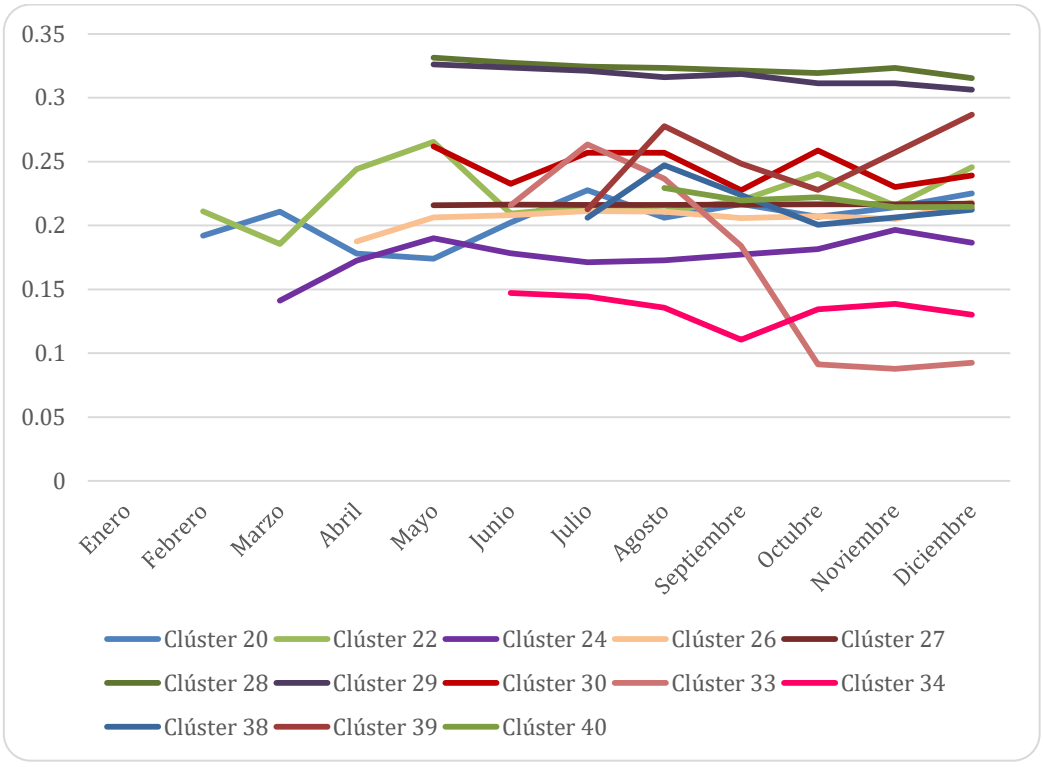

Figura 9. Evolución del centroide de los últimos grupos con LAMDA

Comparación de las Técnicas

Por último, en la Tabla 1 se presenta una comparación de las métricas de ambas ejecuciones. La diferencia de los valores de las métricas entre ambas ejecuciones, posiblemente se deba a la forma como se calcularon. En X-means se calcularon automáticamente con el resultado final de la ejecución. En el caso de LAMDA no se pudo realizar usando las librerías

existentes (es decir, fue realizado por nosotros). Por otro lado, cabe recordar que los datos para LAMDA están normalizados para realizar el agrupamiento, por lo tanto, los valores están en magnitudes diferentes, lo cual termina penalizando a LAMDA al hacer los cálculos.

	X-means		LAMDA	
	Silueta	Davies-Boulding	Silueta	Davies-Boulding
Enero	0,446	0,623	0,06947	102,205
Febrero	0,388	0,645	0,1216	14,496
Marzo	0,389	0,604	0,0143	88,878
Abril	0,384	0,614	0,0419	42,038
Mayo	0,346	0,589	0,1633	15,117
Junio	0,39	0,598	0,132	21,2
Julio	0,387	0,626	0,1617	22,321
Agosto	0,382	0,614	0,0912	27,648
Septiembre	0,377	0,597	0,2152	13,423
Octubre	0,386	0,603	0,284	35,248
Noviembre	0,384	0,638	0,2192	24,321
Diciembre	0,381	0,599	0,161	31,294

Tabla 1. Resultados de las técnicas de agrupamiento

A modo de ejemplo, realizaremos el cálculo de la puntuación de Davies-Bouldin para LAMDA. Recordamos que la fórmula para este cálculo es la indicada en la ecuación 11. Tomemos como ejemplo los 2 primeros clusters del mes de enero donde i será el primer clúster y j el segundo. Para todos los cálculos de distancia, se ha utilizado la distancia euclídea.

$$s_i = 0.016243 \qquad s_j = 0.029149 \qquad d_{ij} = 0.016145$$

Con estos datos, el valor de $R_{ij} = 2.81149$. Este cálculo se realiza comparando cada clúster con todos sus vecinos, seleccionando el resultado máximo para cada clúster. Para ese mes da una puntuación bastante alta dado que hay 2 clústeres bastante cercanos entre sí. Las puntuaciones de cada clúster se detallan en la Tabla 2.

4,1530027	14,76798
19,227854	109,6097
19,227854	14,76798
486,04616	109,6097
486,04616	8,7164474
33,077299	15,055232
8,3582875	

Tabla 2. Puntuación para Davies- Bouldin de los clústeres de enero

Ahora bien, algunas cosas podemos concluir al compararlas. Según las métricas, X-means da consistentemente mejores resultados en ambas métricas. Por otro lado, el coeficiente de Silueta es una excelente métrica en datos con comportamiento espacial circular, mientras que Davies- Bouldin es mejor en los otros casos. Según los resultados obtenidos, se podría intuir que la distribución espacial de los datos es circular, por lo que silueta sería la mejor métrica para compararlas. Ahora bien, X-Means no cambia los clústeres, mientras que LAMDA va ajustando el número de clústeres en el tiempo. Esa evolución nos interesa estudiar.

Análisis de la evolución de los clústeres

Comparando la evolución de ambos algoritmos, al final de la ejecución han quedado 20 y 26 clústeres para X-means y LAMDA, respectivamente. En este apartado, analizaremos la evolución de los clústeres de LAMDA, ya que presenta un comportamiento más dinámico, creando y fusionando clústeres a lo largo de la ejecución.

Empezaremos analizando la creación y fusión de clústeres mostrada en la Tabla 3. Recordemos que el proceso de agrupamiento en línea es de tipo acumulativo, es decir, se toma en cuenta el comportamiento del mes anterior. La Tabla 3 muestra el mes de referencia, el identificador de los clústeres formados, total de clústeres formados al momento, y también, comentarios donde se menciona si existe fusión de algunos clústeres, así como también, el número de clústeres que se agregan en el mes de referencia.

Inicialmente se forman 16 clústeres, de los cuales, los clústeres identificados con los números 10, 11 y 13 se fusionan con otros clústeres, quedando un total de 13 en el primer mes. Para el segundo mes, se observa que aparte de los 13 clústeres creados el mes anterior, **inicialmente** se agregan 6 clústeres nuevos, de los cuales, los clústeres con id 17, 8, 19 y 21 se fusionan, quedando un total de 15 clústeres.

Mes	id de clústeres creados	Total de clústeres	Comentarios
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16	13	16 clústeres formados y fusiona 10, 11 y 13
2	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22	15	Se forman 6 clústeres adicionales y se fusionan los clústeres 17, 18, 19 y 21
3	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24	16	Se forman 2 clústeres adicionales y se fusiona el clúster 23
4	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26	17	Se forman 2 clústeres adicionales y se fusiona el clúster 25
5	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26, 27, 28, 29, 30	21	Se forman 4 clústeres adicionales y no existe fusión
6	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26, 27,28, 29, 30, 33, 34	23	Se forman 4 clústeres adicionales y fusiona 31 y 32
7	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26, 27, 28, 29, 30, 33, 34, 38, 39	25	Se forman 5 clústeres adicionales y fusiona 35 36 y 37
8	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26, 27, 28, 29, 30, 33, 34, 38, 39, 40	26	Se forma 1 clúster adicionales y no hay fusión
9	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26, 27, 28, 29, 30, 33, 34, 38, 39, 40	26	No hay formación adicional de clústeres

10	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26, 27, 28, 29, 30, 33, 34, 38, 39, 40	26	No hay formación adicional de clústeres
11	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26, 27, 28, 29, 30, 33, 34, 38, 39, 40	26	No hay formación adicional de clústeres
12	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 20, 22, 24, 26, 27, 28, 29, 30, 33, 34, 38, 39, 40	26	No hay formación adicional de clústeres

Tabla 3. Creación y fusión de clústeres con LAMDA

Continuando con el análisis, los 9 primeros clústeres generados en enero se mantienen durante todo el periodo de estudio. Este comportamiento de crear y fusionar clústeres se mantiene hasta septiembre, llegando a generar hasta 40 clústeres, de los cuales permanecen 26. A partir de septiembre no se crean nuevos clústeres ni existen nuevas fusiones, tal que todas las nuevas observaciones/individuos se añaden a uno de los 26 clústeres formados hasta el momento.

Análisis de la Evolución Trimestral

Analicemos la evolución de los clústeres por trimestre. Podemos ver en la Figura 10 como la tendencia general es a mantenerse estable y a seguir una tendencia predecible. Se deja de ver el comportamiento especialmente errático que aparecía en los clústeres 19 y 20 en la Figura 6. Estos clústeres tienen pocos individuos en comparación con el resto de los grupos, lo cual los hace más volátiles a pequeños cambios o a nuevas inclusiones en el clúster.

Siguiendo con el análisis de clústeres, en diciembre podemos ver como varios centroides se encuentran cercanos. Por ejemplo, alrededor de la marca de 0.28 encontramos los clústeres 16 y 39, este último fue creado en julio. El clúster 39 tiene un patrón de

comportamiento similar al del 16, por lo que, con el paso del tiempo, si mantiene esta tendencia es posible que se lleguen a unificar. De la misma forma, podemos estudiar el comportamiento de 3 clústeres que se acercan en diciembre, estos son los clústeres 26, 27 y 40, que quedan agrupados por debajo a la marca de 0,22. Sin embargo, este caso es distinto al anterior, ya que únicamente se aproximan al final del análisis, como podemos ver en la Figura 10. En este caso, habría que estar pendiente a su evolución ya que los 3 vienen con trayectorias diferentes. La ventaja de LAMDA, es que se encarga de comprobar la necesidad de fusión y creación de nuevos clústeres de forma automática.

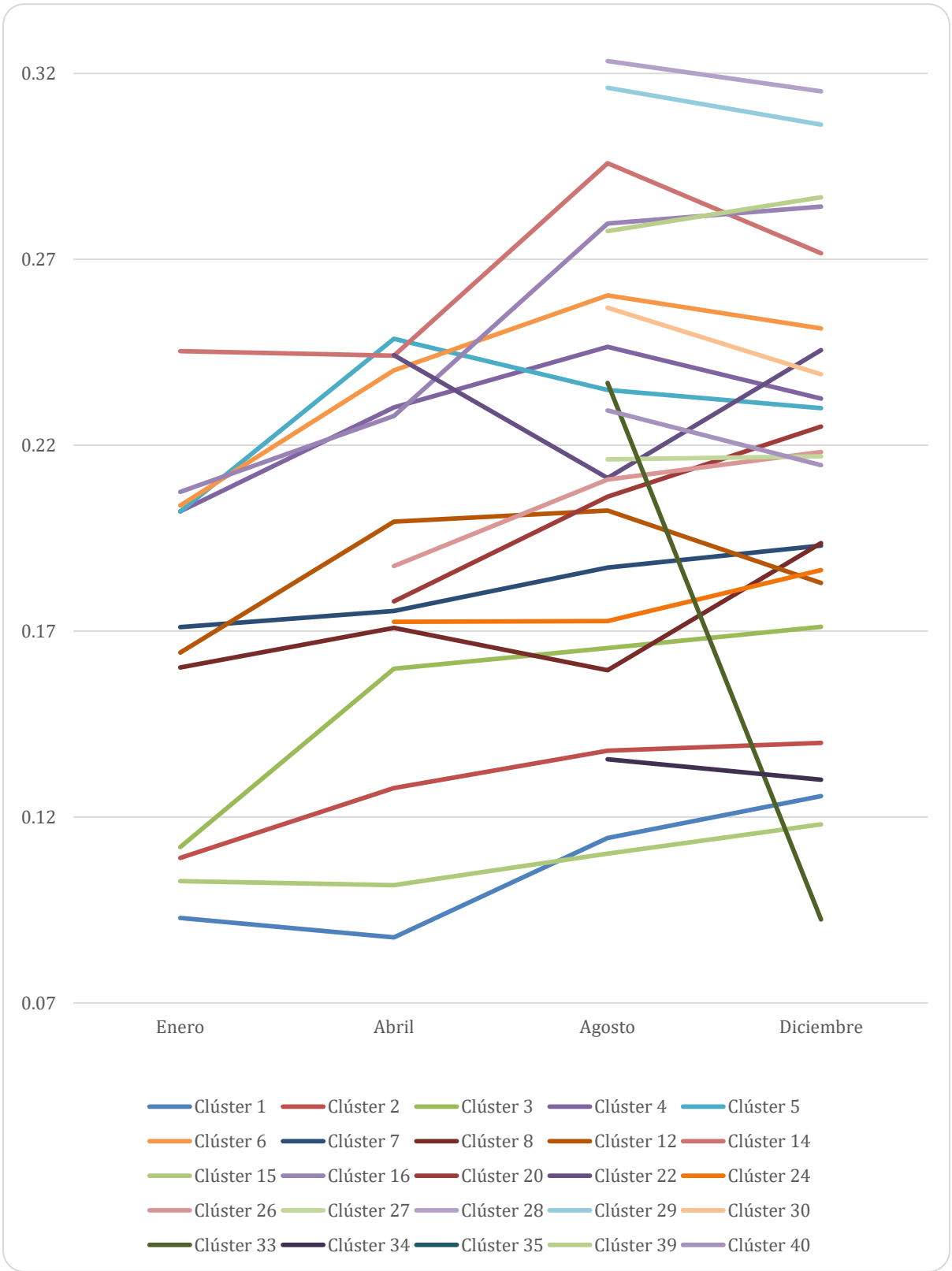

Figura 10. Evaluación de los clústeres por trimestre con LAMDA

Por otro lado, los clústeres más poblados son el 15, 33 y 34, como se puede ver en la Figura 11. Los grupos 15 y 34 son bastante poblados a lo largo de toda su existencia, y el 33 pasa de ser un clúster con pocos individuos a ser uno con mucho peso. Este cambio ocurre cuando, estando en el rango de 0.24 en el mes de agosto, sufre una caída hasta 0.09, donde se estabiliza. En estos clústeres se puede apreciar que su trayectoria (una vez cuentan con un alto número de elementos) es más estable, y únicamente se realizan pequeñas correcciones a sus centroides según se van agregando individuos a los grupos. Vemos entonces que la mayoría de los individuos se encuentran en estos 3 grupos. Particularmente, en diciembre, sus centroides son 15 = 0.170, 33 = 0.092 y 34 = 0.119 (ver Figura 10).

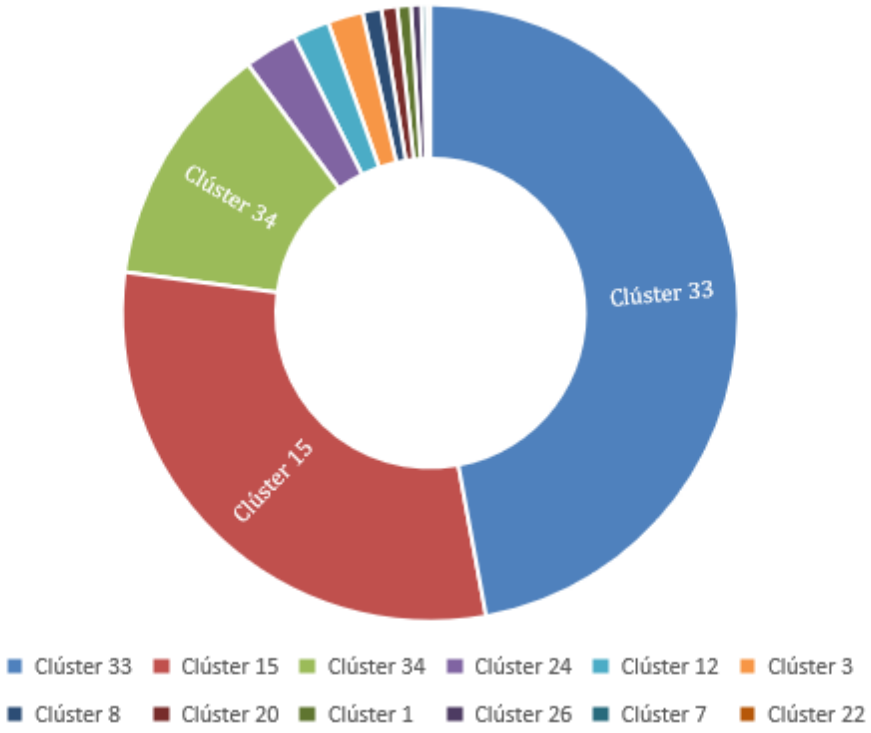

Figura 11. Vista parcial de la distribución de elementos por clúster

En la Figura 11 vemos como la gran parte de los elementos/individuos se han asignado en esos 3 clústeres. Entre ellos ocupan casi un 90% de la ocupación de datos, el resto de los datos se encuentran en los 23 clústeres restantes. Con esto podemos ver que la mayoría de los elementos están en el umbral bajo de consumo, el centroide con el valor más alto de este trio de clústeres es el del clúster 15, con 0.17.

Capítulo 4. Conclusión

En este trabajo hemos realizado un agrupamiento no supervisado con algoritmos de agrupamiento en línea. Al usar X-means y LAMDA, somos capaces de delegar la toma de decisiones sobre el número de clústeres a los algoritmos. Esto se vio particularmente más en LAMDA, ya que fue capaz de ir aumentando y/o disminuyendo el número de grupos. En X-means no pudimos ver este comportamiento, ya que, desde la primera iteración creó el número máximo de clústeres. Creemos que esto se debe a la cantidad de descriptores, que permitió que X-means determinará con rapidez que se necesitaba un alto número de clústeres para este problema.

También hemos visto una importante diferencia en el tiempo de computación. Para realizar la agrupación completa X-means únicamente tardó unos minutos, mientras que LAMDA tardó aproximadamente 3 días. Por otro lado, un análisis sin límite de clústeres es más apropiado en un escenario real (por ejemplo, en el caso de X-means se acotaron en un caso los valores de K), independientemente del tiempo que conlleve.

El análisis de los centroides con LAMDA ha dejado claro que la mayoría representan un consumo bajo. También, según la evolución de los mismos se podría estudiar las tendencias de consumo. Por otro lado, con X-means se vio el comportamiento anormal de algunos clústeres (valores atípicos). En menos palabras, el análisis de la evolución de los centroides de los grupos permite realizar procesos de tomas de decisión en el mundo energético más preciso (meses de mayor consumo, comportamiento anormales, etc.).

Algunos aspectos a tener en cuenta para futuros trabajos son:

- Tener datos tanto del consumo eléctrico (aplicado en nuestro caso) junto con datos de los usuarios, para favorecer un análisis más completo y especifico del

problema en cuestión (por ejemplo, perfilar el comportamiento energético de un individuo).

- En caso de comparar algoritmos de aprendizaje automático es importante realizar el cálculo de métricas usando la misma librería, para evitar confusión a la hora de estudiar los resultados.

Bibliografía

[1] K. Akkaya *et al*, "IoT-based occupancy monitoring techniques for energy-efficient smart buildings," in - *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2015. DOI: 10.1109/WCNCW.2015.7122529.

[2] K. Patel et al, Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing*, vol. 6, (5), pp. 6122-6131, 2016.

[3] C. Gray *et al*, "'Smart' Is Not Free: Energy Consumption of Consumer Home Automation Systems," *IEEE Transactions on Consumer Electronics*, vol. 66, *(1)*, pp. 87-95, 2020. DOI: 10.1109/TCE.2019.2962605.

[4] R. Yang and L. Wang, "Development of multi-agent system for building energy and comfort management based on occupant behaviors," *Energy Build.*, vol. 56, pp. 1-7, 2013. DOI: 10.1016/j.enbuild.2012.10.025.

[5] E. Fotopoulou *et al*, "Providing Personalized Energy Management and Awareness Services for Energy Efficiency in Smart Buildings," *Sensors*, vol. 17, *(9)*, pp. 2054, 2017. DOI: 10.3390/s17092054.

[6] J. S. Saltz, "CRISP-DM for data science: Strengths, weaknesses and potential next steps," in *IEEE International Conference on Big Data (Big Data)*, 2021. DOI: 10.1109/BigData52589.2021.9671634.

[7] R. Wirth and J. Hipp, "CRISP-DM: Towards a standard process model for data mining," in in *4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 2000.

[8] G. Yoon *et al*, "Prediction of machine learning base for efficient use of energy infrastructure in smart city," in *International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pp. 32-35, 2019.

[9] Q. Xiao *et al*, "Energy Efficiency Modeling for Configuration-Dependent Machining via Machine Learning: A Comparative Study," *Tase*, vol. 18, *(2)*, pp. 717-730, 2021. DOI: 10.1109/TASE.2019.2961714.

[10] Z. Wu and W. Chu, "Sampling strategy analysis of machine learning models for energy consumption prediction," in IEEE 9th International Conference on Smart Energy Grid Engineering, pp. 77-81, 2021, doi: 10.1109/SEGE52446.2021.9534987.

[11] O. A. Olanrewaju, "Predicting industrial sector's energy consumption: Application of support vector machine," IEEE International Conference on Industrial Engineering and Engineering Management pp. 1597-1600, 2019, doi: 10.1109/IEEM44572.2019.8978604.

[12] W. Chu *et al*, "Effects of wiring density and pillar structure on chip package interaction for advanced cu low-k chips," in IEEE International Reliability Physics Symposium, 2020, doi: 10.1109/IRPS45951.2020.9128333.

[13] D. N. Darlis *et al*, "Random forest approach for energy consumption behavior analysis," in IEEE Symposium on Industrial Electronics & Applications 2020, doi: 10.1109/ISIEA49364.2020.9188072.

[14] N. Zhang and D. Shetty, "An effective LS-SVM-based approach for surface roughness prediction in machined surfaces," *Neurocomputing (Amsterdam)*, vol. 198, pp. 35-39, 2016. DOI: 10.1016/j.neucom.2015.08.124.

[15] A. M. Abdulshahed *et al*, "Thermal error modelling of a gantry-type 5-axis machine tool using a Grey Neural Network Model," *Journal of Manufacturing Systems*, vol. 41, pp. 130-142, 2016. DOI: 10.1016/j.jmsy.2016.08.006.

[16] D. Kong, Y. Chen and N. Li, "Gaussian process regression for tool wear prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 556-574, 2018. DOI: 10.1016/j.ymssp.2017.11.021.

[17] D. A. Bashawyah and S. M. Qaisar, "Machine learning based short-term load forecasting for smart meter energy consumption data in london households," in IEEE 12th International Conference on Electronics and Information Technologies, pp. 99-102, 2021, doi: 10.1109/ELIT53502.2021.9501104., 2021, pp. 99-102.

[18] M. Paluszek and S. Thomas, *MATLAB Machine Learning Recipes.* Berkeley, CA: Apress L. P, 2019.

[19] R. Gandhi. (). *Support Vector Machine — Introduction to Machine Learning Algorithms*. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[20] J. Aguilar *et al*, "Traceability analysis of patterns using clustering techniques," in *Advances in Artificial Intelligence and Applied Cognitive Computing*Anonymous Cham: Springer International Publishing, 2021, pp. 235-250.

[21] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python.* (First edition ed.) 2016.

[22] A. M. Bagirov, N. Karmitsa and S. Taheri, *Partitional Clustering Via Nonsmooth Optimization.* (1st ed. 2020. ed.) Cham: Springer International Publishing, 2020.

[23] J. Han, J. Pei and M. Kamber, *Data Mining.* Saint Louis: Elsevier Science & Technology, 2011.

[24] C. C. Aggarwal and C. K. Reddy, *Data Clustering.* (1st ed.) Philadelphia, PA: CRC Press, 2014*31*.

[25] H. Späth, *Cluster Analysis Algorithms for Data Reduction and Classification of Objects.* Chichester: Horwood, 1980*4*.

[26] L. Rokach and O. Maimon, "Clustering methods," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2005, pp. 321-352.

[27] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy.* San Francisco: Freeman, 1973.

[28] B. King, "Step-Wise Clustering Procedures," *Journal of the American Statistical Association*, vol. 62, *(317)*, pp. 86-101, 1967. DOI: 10.1080/01621459.1967.10482890.

[29] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Computer Journal*, vol. 26, *(4)*, pp. 354-359, 1983. DOI: 10.1093/comjnl/26.4.354.

[30] R. M. Aliguliyev, "Performance evaluation of density-based clustering methods," *Information Sciences*, vol. 179, *(20)*, pp. 3583-3602, 2009. DOI: 10.1016/j.ins.2009.06.012.

[31] J. D. BANFIELD and A. E. RAFTERY, "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, vol. 49, *(3)*, pp. 803-821, 1993. DOI: 10.2307/2532201.

[32] M. Ester *et al*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Dec 31, 1996,

[33] B. B. Chaudhuri and G. Garai, "Grid Clustering With Genetic Algorithm and Tabu Search Process," *Journal of Pattern Recognition Research*, vol. 4, *(1)*, pp. 152-168, 2009. DOI: 10.13176/11.125.

[34] L. Morales and J. Aguilar, "An Automatic Merge Technique to Improve the Clustering Quality Performed by LAMDA," *IEEE Access*, vol. 8, pp. 162917-162944, 2020. DOI: 10.1109/ACCESS.2020.3021675.

[35] D. Pham, S. Dimov and C. Nguyen, "Selection of K in K -means clustering," *Institution of Mechanical Engineers Part C-Journal of Mechanical Engineering Science - PROC INST MECH ENG C-J MECH E*, vol. 219, pp. 103-119, 2005. DOI: 10.1243/095440605X8298.

[36] D. Pelleg and A. Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," *Machine Learning, P*, 2002.

[37] C. Quintero-Gull and J. Aguilar, "LAMDA-HSCC: A semi-supervised learning algorithm based on the multivariate data analysis," *Expert Systems with Applications*, vol. 202, pp. 117479, 2022. DOI: 10.1016/j.eswa.2022.117479.

[38] M. Mizumoto, "Pictorial representations of fuzzy connectives, Part I: Cases of t-norms, t-conorms and averaging operators," *Fuzzy Sets and Systems*, vol. 31, *(2)*, pp. 217-242, 1989. DOI: 10.1016/0165-0114(89)90005-5.

[39] F. A. Ruiz *et al*, "A new criterion to validate and improve the classification process of LAMDA algorithm applied to diesel engines," *Engineering Applications of Artificial Intelligence,* vol. 60, pp. 117-127, 2017. DOI: 10.1016/j.engappai.2017.02.005.

[40] C. Bedoya, C. Uribe and C. Isaza, "Unsupervised feature selection based on fuzzy clustering for fault detection of the tennessee eastman process," in *Advances in Artificial Intelligence – IBERAMIA 2012*, Heidelberg: Springer Berlin Heidelberg, pp. 350-360.

[41] M. Royapoor *et al*, "Building as a virtual power plant, magnitude and persistence of deferrable loads and human comfort implications," *Energy and Buildings,* vol. 213, pp. 109794, 2020. DOI: 10.1016/j.enbuild.2020.109794.

[42] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics,* vol. 20, pp. 53-65, 1987. DOI: 10.1016/0377-0427(87)90125-7.

[43] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *Tpami,* vol. PAMI-1, *(2),* pp. 224-227, 1979. DOI: 10.1109/TPAMI.1979.4766909.

[44] A. Novikov, "PyClustering: Data Mining Library," *Journal of Open Source Software,* vol. 4, *(36),* pp. 1230, 2019. DOI: 10.21105/joss.01230.

Universidad de Alcalá

Escuela Politécnica Superior

ESCUELA POLITECNICA SUPERIOR