Métodos de clústering para series de tiempo

Orlando Uc

11 de junio de 2020

Contenido

Motivación

- **Importancia:** El análisis de series de tiempo es un área de la estadística, permite modelar datos con una dependencia temporal, y realizar pronósticos precisos y confiables. Los métodos de machine learning han ampliado el alcance de las series de tiempo, al permitir analizar un mayor número de series desde una perspectiva moderna.

- **Objetivo:** Introducir y ejemplificar la aplicación de métodos de clústering para series de tiempo, en particular: Euclidian k-means, Soft DTW k-means, DBA k-means y GAK k-means.

Introducción a series de tiempo

Definición

Un proceso estocástico es una colección de variables aleatorias $\{X_t\}_{t \in T}$, donde T es un conjunto de índices arbitrario.

En palabras muy sencillas, una serie de tiempo es una colección de observaciones de un fenómeno en particular, indexada a través del tiempo.

Definición

Una serie de tiempo es una realización de un proceso estocástico en tiempo discreto, donde los elementos del conjunto de índices T están ordenados.

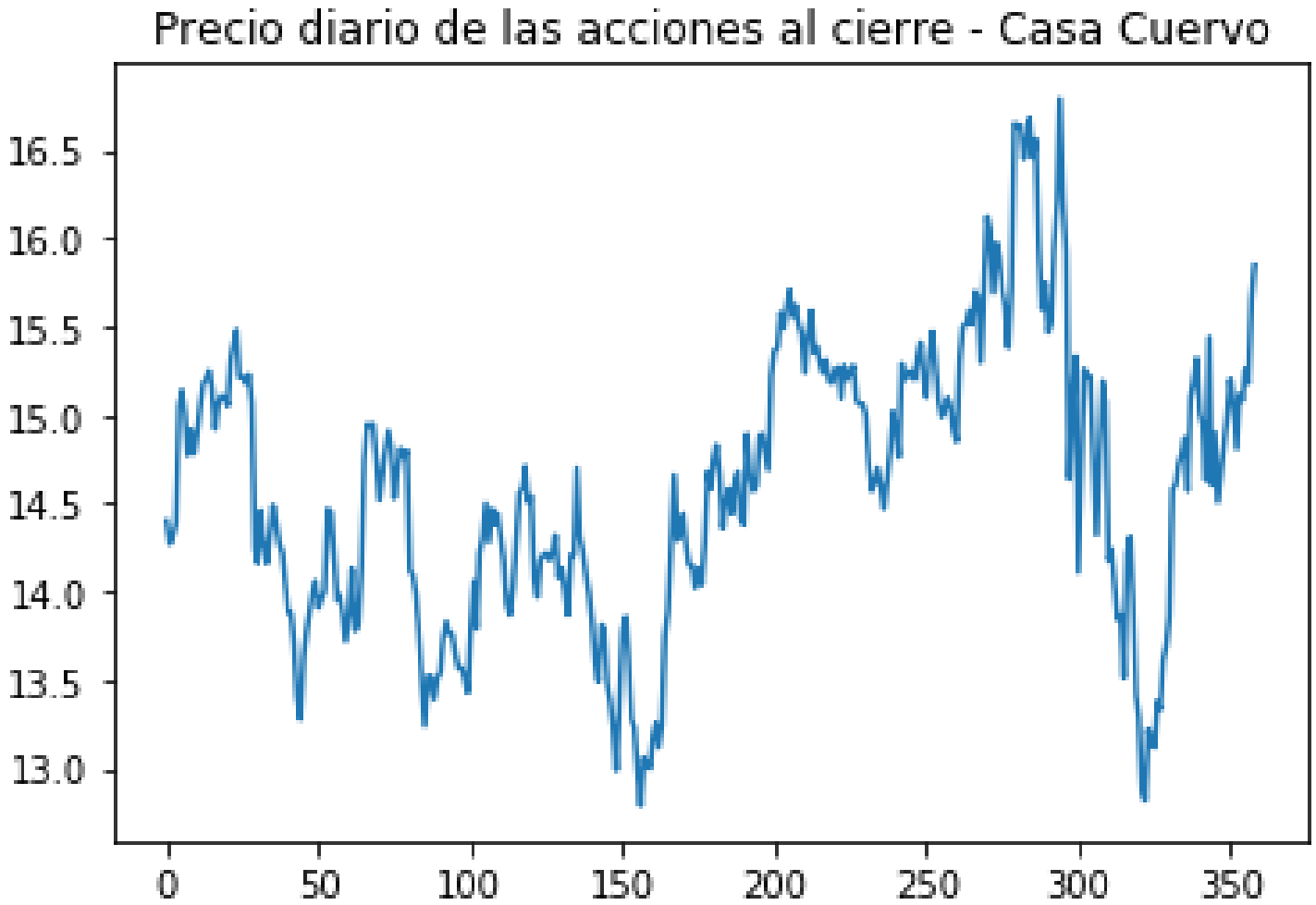

Figura: Precio diario al cierre en MXN de las acciones de Casa Cuervo entre el 1/enero/2019 y el 8/mayo/2020.

Dynamic Time Warping (DTW)

Sean $X_t = \{x_1, x_2, \cdots, x_n\}$ y $Y_t = \{y_1, y_2, \cdots, y_n\}$ dos series de tiempo. Para poder hacer comparaciones entre ellas primero es necesario definir una medida de distancia adecuada.

La primera opción es aplicar la distancia euclidiana definida como:

$$D(X_t, Y_t) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2} \tag{1}$$

Pero esta distancia no permite contemplar la dependencia temporal entre las series. Además, es una distancia rígida, en el sentido de que las comparaciones son entre las observaciones en el mismo tiempo, lo cual solo permite calcular distancias entre series de la misma longitud.

Dynamic Time Warping (Deformación Dinámica del Tiempo) es un algoritmo que busca la alineación no-lineal óptima entre dos series. Fue introducido por Sakoe (1971) con aplicaciones en reconocimiento de voz.

La alineación óptima puede ser calculada recursivamente por:

$$D(X_i, Y_i) = \delta(x_i, y_i) + \min \left\{ \begin{array}{c} D(x_{i-1}, y_{j-1}) \\ D(x_i, y_{j-1}) \\ D(x_{i-1}, y_j) \end{array} \right\} \tag{2}$$

donde X_i es la subsecuencia $\{x_1, \cdots, x_i\}$ y Y_i es la subsecuencia $\{y_1, \cdots, y_i\}$. La distancia completa está dada por $D(X_t, Y_t) = D(X_n, Y_n)$.

Tiene la ventaja de que puede calcular distancias entre series de diferente longitud.

La implementación directa del algoritmo DTW lleva a un tiempo de cómputo exponencial.

Sin embargo, el hecho de que para obtener la distancia completa $D(X_t, Y_m)$ se tenga que realizar una recursión, permite memorizar las distancias parciales, lo que hace que el cálculo de la alineación óptima un proceso de orden $n \times m$, para una serie de longitud n y otra de dimensión m.

A continuación, el algoritmo DTW tomado de (Petitjean, Ketterlin y Gançarski, 2010).

Algorithm 1 DTW

Require: $A = \{a_1, \cdots, a_S\}$
Require: $B = \{b_1, \cdots, b_T\}$
Let δ be a distance between coordinates of sequences
Let $m[S, T]$ be the matrix of couples (cost,path)

$m[1,1] \leftarrow (\delta(a_1, b_1), (0,0))$
for $i \leftarrow 2$ to S **do**
    $m[i,1] \leftarrow (m[i-1,1,1] + \delta(a_i, b_1), (i-1,1))$
end for
for $j \leftarrow 2$ to T **do**
    $m[1,j] \leftarrow (m[1,j-1,1] + \delta(a_1, b_i), (1,j-1))$
end for
for $i \leftarrow 2$ to S **do**
    for $j \leftarrow 2$ to S **do**
        $minimum \leftarrow minVal(m[i-1,j], m[i,j-1], m[i-1,j-1])$
        $m[i,j] \leftarrow (first(minimum) + \delta(a_i, b_j), second(minimum))$
    end for
end for
return $m[S, T]$

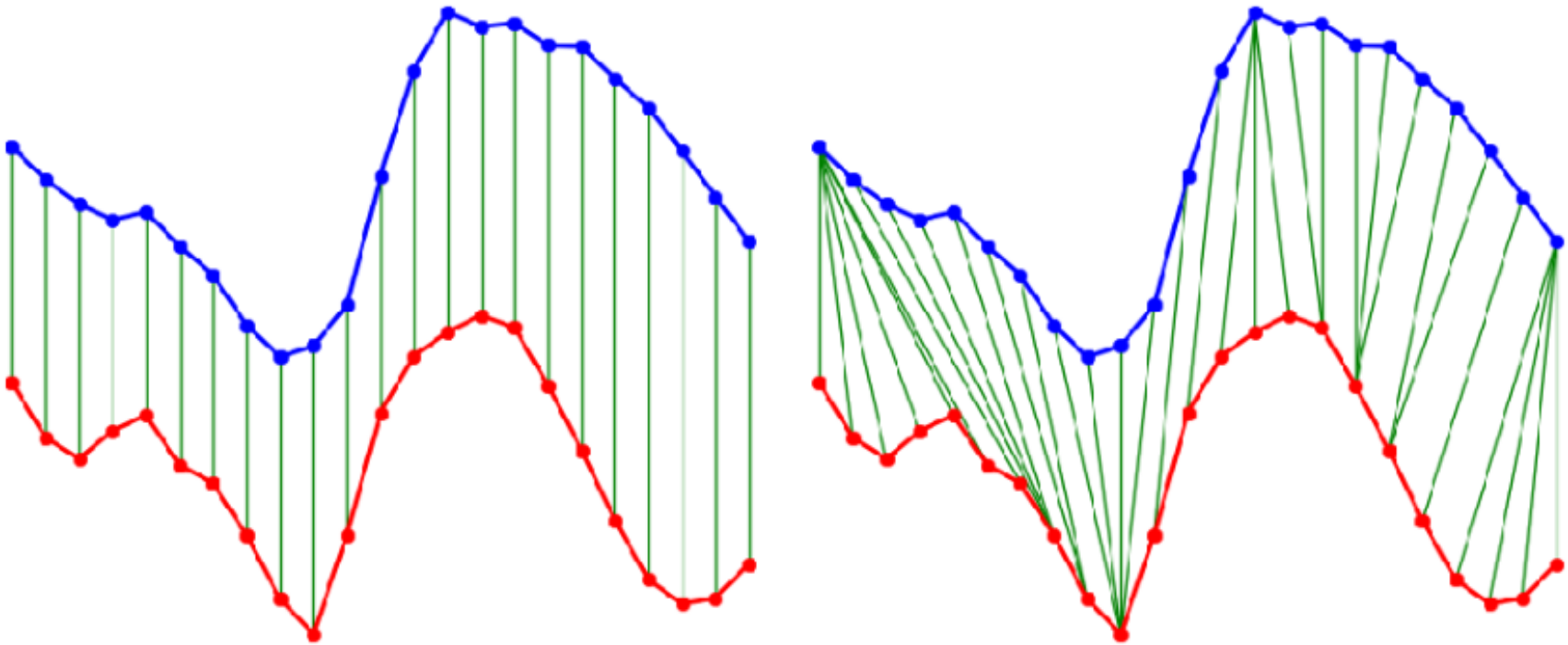

Figura: Comparación del cálculo de la distancia euclidiana contra la distancia DTW (Zhang, Tang, Huo y Zhou, 2014) .

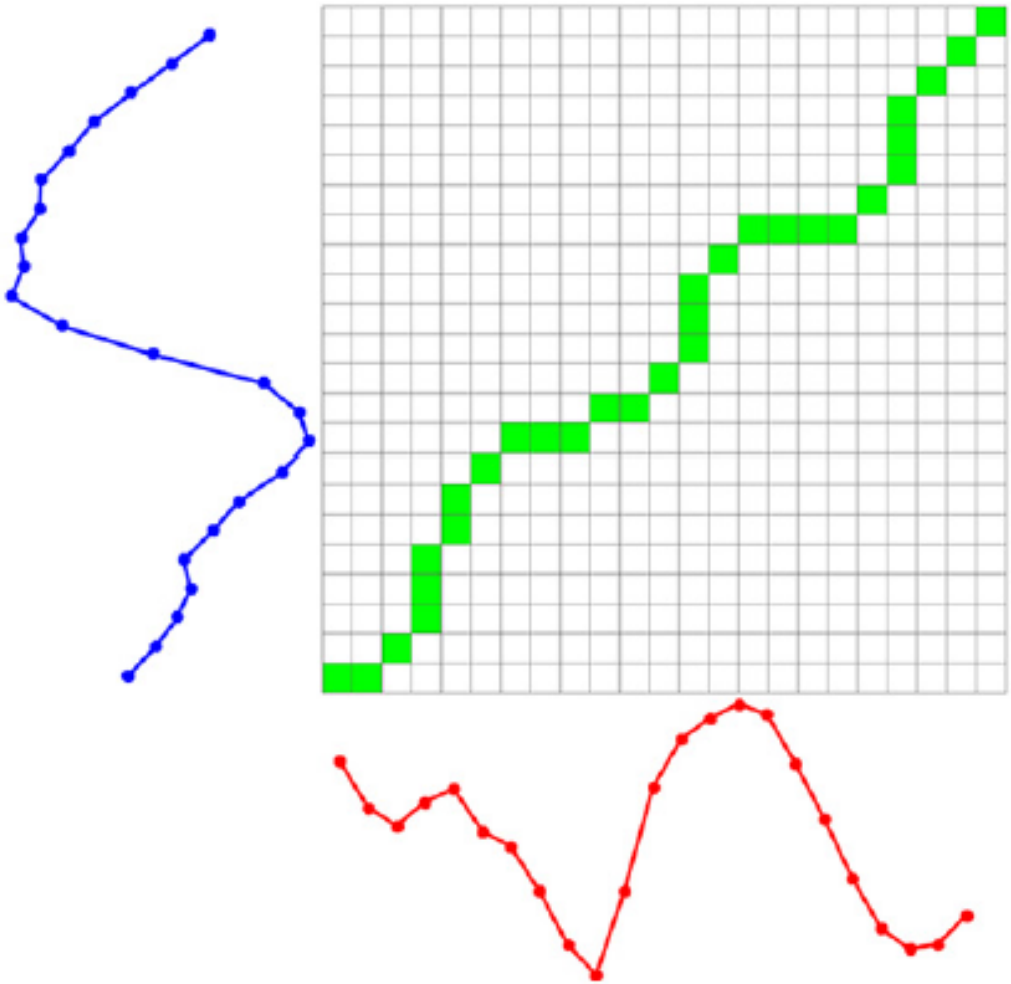

Figura: Ejemplo de la matriz del cálculo de la distancia DTW (Zhang, et al., 2014).

Métodos de clústering para series de tiempo

En general, los métodos de clústering para series de tiempo se pueden agrupar en:

1. **Clústering de series de tiempo basado en disimilaridades:** Están basados en calcular distancias entre las series.
2. **Clústering de series de tiempo basado en modelos:** El objetivo es identificar similitudes entre los modelos propuestos para cada serie.
3. **Clústering de series de tiempo basado en dependencias.** Se basan en cuantificar las correlaciones entre las series.

(Liao, 2005) y (Aghabozorgi, Shirkhorshidi y Wah, 2015).

Euclidian k-means

Un método introductorio para clústering de series de tiempo es Euclidian k-means. El objetivo es, dados K centroides elegidos al azar, asignar cada serie al clúster cuya distancia euclidiana sea la mínima.

A continuación, una adaptación del algoritmo tomado de (Bo, Luo y Vo, 2016).

Algorithm 2 Euclidian k-means

Require: Number of clusters K
Require: Data set $T = \{t_1, t_2, \cdots, t_n\}$ con $t_i = (x_i, y_i)$

1: Choose randomly K instances to be the m_k initial centroids.
2: For each instance, assign it to the cluster the centroid of which is the closest to the instance.

$$\min_{1 \leq k \leq K} ||t_i - m_k||^2$$

3: For each cluster, recompute its centroid based on the instances in that cluster.
4: If the convergence criterion is satisfied, then stop; otherwise, go back to Step 2.

Soft k-means

Una variante del algoritmo k-means es Soft (Fuzzy) k-means.

La diferencia radica en que en lugar de minimizar la distancia euclidiana entre cada serie y los centroides, se considera una distancia ponderada por la plausibilidad de que la i-ésima serie pertenezca al k-ésimo clúster. El cálculo de los centroides también es ponderado.

Si en lugar de la distancia euclidiana se considera la distancia DTW el método se llama Soft DTW k-means.

El algoritmo original fue propuesto por Kaufman y Rousseeuw (1990).

Algorithm 3 Soft DTW k-means

Require: Number of clusters K
Require: Fuzzy parameter q
Require: Data set $T = \{t_1, t_2, \cdots, t_n\}$ con $t_i = (x_i, y_i)$

1: Choose randomly K instances to be the m_k initial centroids.
2: For each instance, assign it to the cluster the centroid of which is the closest to the instance.

$$\min_{1 \leq k \leq K} \gamma_{ik}^{q} DTW(t_i, m_k)^2$$

with

$$\gamma_{ik}^{q} = \frac{1}{\sum_{j=1}^{K} \left(\frac{DTW(t_i, m_k)}{DTW(t_i, m_j)} \right)^{\frac{2}{1}}}$$

$$m_k = \frac{\sum_{i=1}^{n} \gamma_{ik}^{q} t_i}{\sum_{i=1}^{n} \gamma_{ik}^{q}}$$

$$\sum_{k} \gamma_{ik} = 1$$

3: For each cluster, recompute its centroid based on the instances in that cluster.
4: If the convergence criterion is satisfied, then stop; otherwise, go back to Step 2.

DBA k-means

DTW Barycenter Averaging (DBA) es uno de los métodos más populares para clústering de series de tiempo. Fue propuesto por Petitjean, Ketterlin y Gançarski (2010).

El objetivo es minimizar la suma de cuadrados de las distancias DTW de una serie promedio con respecto al conjunto de series.

DBA sigue un esquema Expectation-Maximization. Se define una serie promedio $\overline{T}$ e iterativamente:

1. Se considera a $\overline{T}$ fija y se calcula la distancia DTW entre cada serie y $\overline{T}$, para poder encontrar el mejor alineamiento múltiple M.
2. Ahora se se considera fijo a M y se actualiza $\overline{T}$ como el mejor baricentro consistente con M.

El concepto de baricentro es en el sentido interpretativo de que el algoritmo calcula un promedio ponderado de las distancias DTW entre las series.

A continuación, el algoritmo DBA propuesto en (Petitjean, Forestier, Webb, Nicholson, Chen y Keogh, 2014).

Algorithm 4 DBA

Require: $T = \{t_1, t_2, \cdots, t_n\}$ con $t_i = (x_i, y_i)$, the set of sequences to average.
Require: I, the number of iterations
1: $\overline{T}$ el promedio ponderado inicial.
2: **for** $i \leftarrow 1$ to I **do**
3: $\quad \overline{T} = \text{DBA\_update}(\overline{T}, M)$
4: **end for**
5: **return** $\overline{T}$

Algorithm 5 DBA_update

Require: $\overline{T_{init}}$, the average sequence to refine (of length L).
Require: $T = \{t_1, t_2, \cdots, t_n\}$ con $t_i = (x_i, y_i)$, the set of sequences to average.

1: *// Step #1: compute the multiple alignment for* $\overline{T_{init}}$
2: alignment $= [0, \cdots, 0]$ *// array of L empty sets*
3: **for** each S in T **do**
4: alignment_for_S = DTW_multiple_alignment $(\overline{T_{init}}, S)$
5: **for** i=1 to L **do**
6: alignment[i] = alignment[i] $\cup$ alignment_for_S[i]
7: **end for**
8: **end for**
9: *// Step #2: compute the multiple alignment for the alignment*
10: $\overline{T}$ be a sequence of length L
11: **for** i=1 to L do **do**
12: $\overline{T}(i) =$ *mean* (alignment[i]) *// arithmetic mean on the set*
13: **end for**
14: **return** $\overline{T}$

GAK k-means

La distancia DTW puede ser generalizada al concepto *alineamiento* entre series de tiempo.

Sean $X_t = (x_1, \cdots, x_n)$ y $Y_t = (y_1, \cdots, y_m)$, entonces un alineamiento π entre las series es un par de vectores (π_1, π_2) de tamaño $p \leq n + m - 1$ tal que $1 = \pi_1(1) \leq \cdots \leq \pi_1(p) = n$ y $1 = \pi_2(1) \leq \cdots \leq \pi_2(p) = m$, con incrementos unitarios y sin repeticiones simultáneas.

Las dos coordenadas π_1 y π_2 del alineamiento π son conocidas como funciones deformadoras.

Sea $A(n, m)$ el conjunto de todos los alineamientos entre dos series de longitud n y m, entonces la distancia DTW puede ser definida como:

$$DTW(x, y) = \min_{\pi \in A(n,m)} D_{x,y}(\pi) \tag{3}$$

donde, si se define $|\pi|$ como la longitud de π, entonces:

$$D_{x,y}(\pi) = \sum_{i=1}^{|\pi|} \varphi(x_{\pi_1}(i), y_{\pi_2}(i)) \tag{4}$$

El Global Alignment Kernel (Cuturi, 2011) es definido como el mínimo suavizado exponencial de todas las distancias de alineamiento

$$K_{GA}(x, y) = \sum_{\pi \in A(n.m)} e^{-D_{x,y}(\pi)} \tag{5}$$

La ecuación 21 se puede reescribir usando la función de similitud k inducida desde la divergencia φ como $k = e^{-\varphi}$:

$$K_{GA}(x, y) = \sum_{\pi \in A(n,m)} \prod_{i=1}^{|\pi|} k\left(x_{\pi_1}(i), y_{\pi_2}(i)\right) \tag{6}$$

A continuación, una adaptación del algoritmo propuesto en (Dhillon, Guan y Kulis, 2004).

Algorithm 6 GAK k-means

Require: Kernel matrix K
Require: Number of clusters K
Require: Data set $T = \{t_1, t_2, \cdots, t_n\}$ con $t_i = (x_i, y_i)$

1: Initialize the k clusters: $C_1^{(0)}, \cdots, C_k^{(0)}$
2: Set $t = 0$
3: For each series t_i, finds its new cluster index as:

$$j^*(t_i) = \min_j K_{GA}(t_i, m_j)$$

4: Compute the uptades clusters as:

$$C_j^{t+1} = \{t_i : j^*(t_i) = j\}$$

5: If not converged, set $t = t + 1$ and go to Step 3; Otherwise, stop.
6: **return** $C_1, \cdots, C_k$: partitioning of the points

Ejemplo de aplicación: Clústering sobre los componentes del IPC

El S&P/BMV IPC es el índice bursátil más importante de la Bolsa Mexicana de Valores (BMV). Es un indicador de la estabilidad del mercado financiero mexicano.

Es un índice ponderado, y tiene como componentes a las 35 empresas más grandes de México, ya sea por capitalización, solvencia y/o estabilidad.

Se creó una base de datos con las observaciones diarias de los precios al cierre de las acciones de las 35 empresas que componen el IPC, en el periodo 1/enero/2019 al 8/junio/2020.

Se aplicaron los métodos Euclidian k-means, Soft DTW k-means, DBA k-means y GAK k-means para ajustar $k = 3$ clústers.

Se utilizó el módulo *tslearn* de Python 3, que es un módulo en el que se pueden aplicar varios métodos de machine learning para series de tiempo.

Conclusiones

- Existen diferentes métodos para clústering de series de tiempo, y en general, se pueden agrupar por basados en disimilaridades, basados en modelos y basados en dependencias.
- DTW permite encontrar la distancia entre dos series contemplando la dependencia temporal entre ellas, y a través de una alineación no-lineal óptima.
- Los métodos Euclidian k-means y Soft k-means son bastante similares.
- DBA k-means utiliza la distancia DTW.
- GAK k-means utiliza Global Alignment Kernel.

- Las compañías mexicanas con mejor desempeño son GRUMAB(GRUMA), WALMEX(Walmart de México), CUERVO (Casa Cuervo), LABB (GenomaLab), ELEKTRA(Elektra), Q(Quálitas).
- Las compañias mexicanas con peor desempeño son OMAB(Grupo Aeropuertuario del Centro Norte), ASURB(Grupo Aeropuertuario del Sureste), GAPB(Grupo Aeropuertuario del Pacífico), GENTERA(Compartamos Banco).

Referencias

- Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering–a decade review. *Information Systems*, 53, 16-38.
- Cuturi, M. (2011). Fast global alignment kernels. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 929-936).
- Dhillon, I. S., Guan, Y., & Kulis, B. (2004, August). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 551-556).
- Kaufman, L. R., & Rousseeuw, P. PJ (1990) *Finding groups in data: An introduction to cluster analysis*. Hoboken NJ John Wiley & Sons Inc, 725.
- Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38(11), 1857-1874.

- Petitjean, F., Forestier, G., Webb, G. I., Nicholson, A. E., Chen, Y., & Keogh, E. (2014, December). Dynamic time warping averaging of time series allows faster and more accurate classification. In *2014 IEEE international conference on data mining* (pp. 470-479). IEEE.
- Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3), 678-693.
- Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1), 43-49.
- Vo, V., Luo, J., & Vo, B. (2016). Time series trend analysis based on K-means and support vector machine. *Computing and Informatics*, 35(1), 111-127.
- Zhang, Z., Tang, P., Huo, L., & Zhou, Z. (2014). MODIS NDVI time series clustering under dynamic time warping. *International Journal of Wavelets, Multiresolution and Information Processing*, 12(05), 1461011.