Introducción a la Ciencia de Datos

Maestría en Ciencias de la Computación

Dr. Irvin Hussein López Nava

Lectura 5: DTW

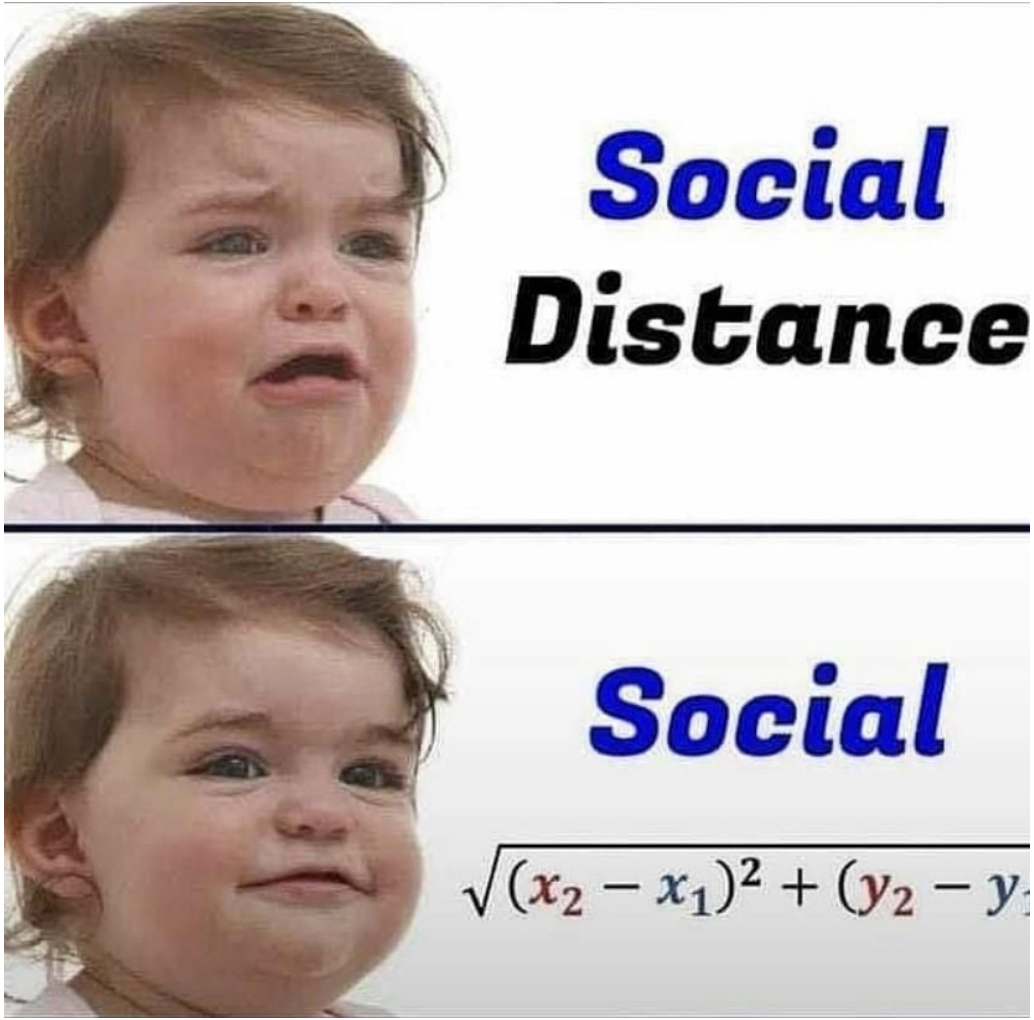
Irvin Hussein Lopez Nava • 16 oct

100 puntos **Fecha de entrega: 23:00**

Preparar un reporte de máximo una cuartilla con un análisis del artículo adjunto. El reporte debe finalizar con una crítica al artículo, el cual será abordado en las siguientes sesiones.

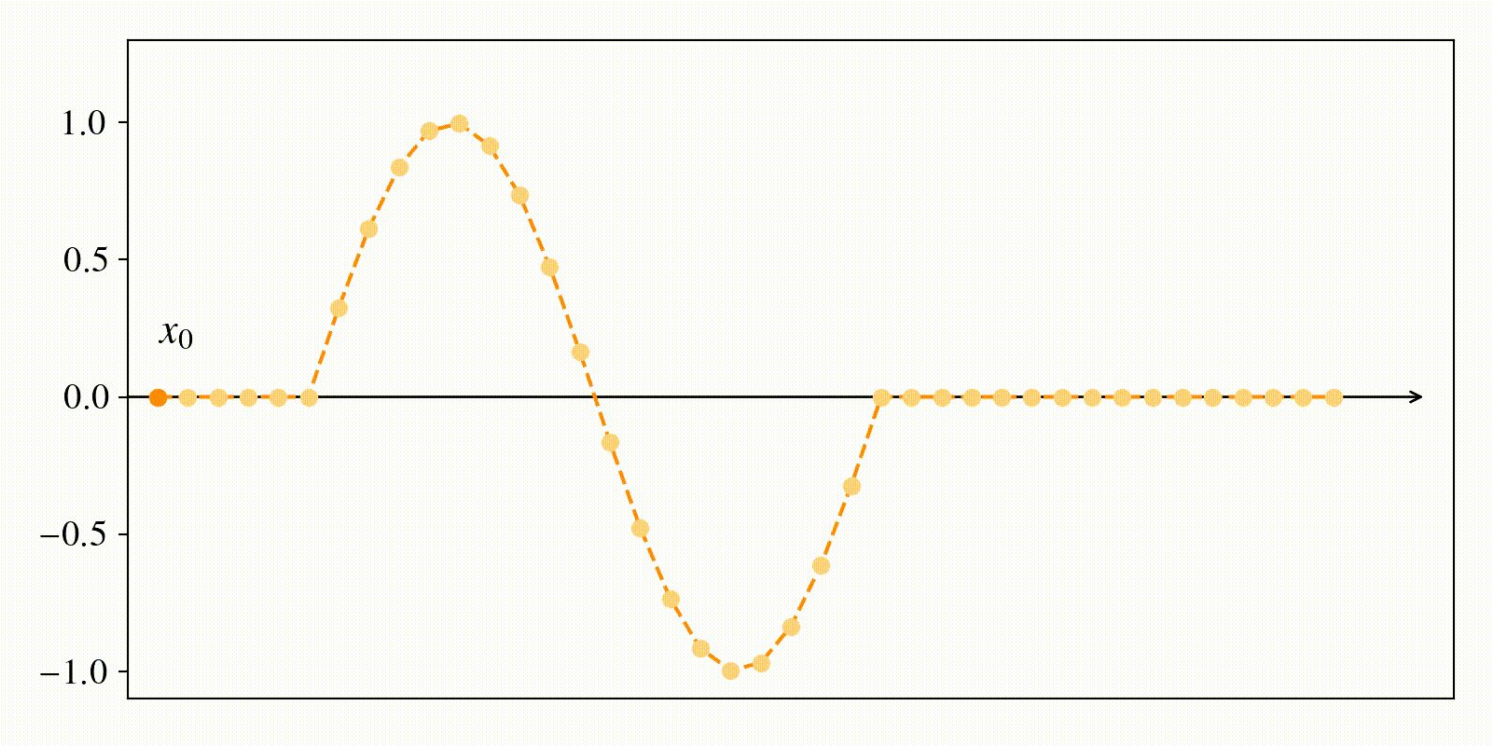
Seto, S., Zhang, W., & Zhou, Y. (2015, December). Multivariate time series classification using dynamic time warping template selection for human activity recognition. In *2015 IEEE symposium series on computational intelligence* (pp. 1399-1406). IEEE.

1512.06747.pdf
PDF


Social
Distance
Social
$\sqrt{(x_2 - x_1)^2 + (y_2 - y$

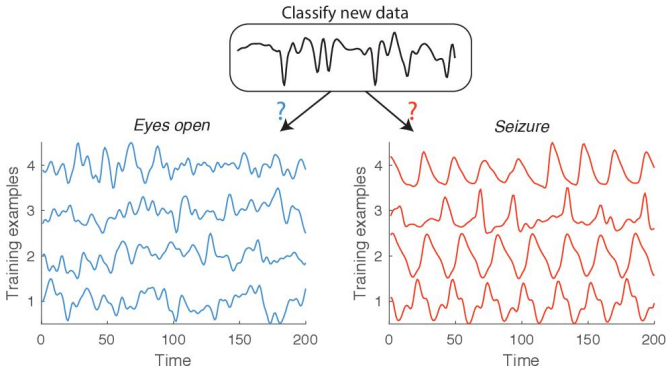
07

Clasificación de series de tiempo


1.0
0.5
x_0
0.0
−0.5
−1.0

Clasificación temporal

- El esquema de clasificación temporal (basado en plantillas) consiste en comparar la **similitud** de las señales entre ellas.
- Se toman como modelos, o plantillas, las señales etiquetadas.
- Se asigna el valor de clase al modelo con mayor similitud.

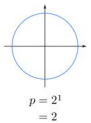


Métricas de distancia

Una forma de comparar series de tiempo es a partir de medidas de distancia. Sean T y S dos series de tiempo de longitud n, y T_i y S_i el ith valor de T y S, respectivamente.

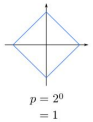
- Distancia Euclidiana (L_2 norm)

$$d(T,S) = \sqrt{\sum_{i=1}^{n} (T_i - S_i)^2}$$



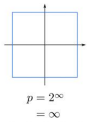
- Distancia Manhattan (L_1 norm)

$$d(T,S) = |T_i - S_i|$$



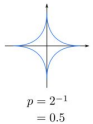
- Distancia Chebyshev (L_∞ norm)

$$d(T,S) = \max_{0<1\leq n} |T_i - S_i|$$



- Distancia Minkowski (L_p norm)

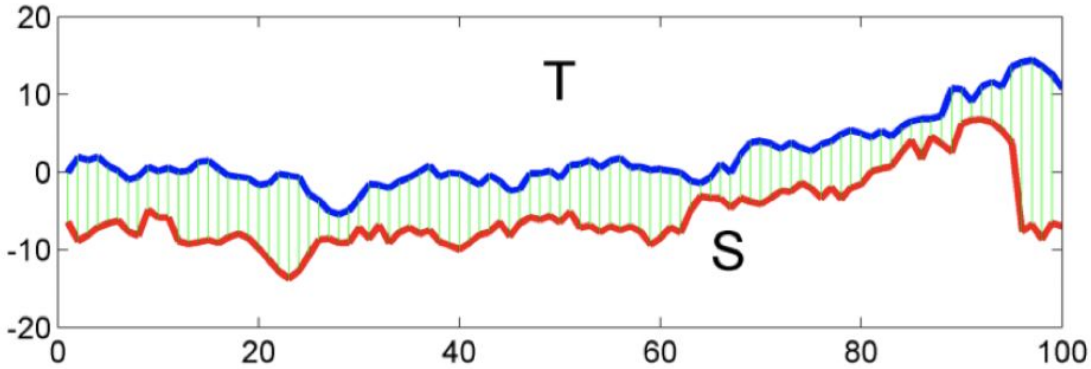
$$d(T,S) = \sqrt[p]{\sum_{i=1}^{n} (T_i - S_i)^p}$$



Distancia Euclidiana

- Es la suma de las distancias cuadradas desde cada punto *n* en una serie de tiempo hasta el punto *n* en la otra.
- Esta distancia y sus variantes presentan varios inconvenientes:
  - Compara solo series de tiempo de la misma longitud.
  - Si dos series de tiempo son idénticas, pero una se desplaza ligeramente a lo largo del eje del tiempo, entonces la distancia euclidiana puede considerar que las señales son muy diferentes entre sí.
  - Presenta problemas con outliers o ruido.
  - Es muy sensible con respecto a las seis transformaciones de señal: desplazamiento, escalado de amplitud uniforme, escalado de tiempo uniforme, escalada bi-uniforme, distorsión de tiempo y escalado de amplitud no uniforme.

Distancia Euclidiana



7.1 Dynamic time warping


¿Qué es el
Alineamiento
dinámico del tiempo?

Definiciones

ChatGPT

Es una técnica utilizada en el procesamiento de señales, reconocimiento de patrones y minería de datos para comparar y alinear secuencias temporales o series de tiempo que pueden variar en velocidad y duración.

https://chat.openai.com/

Wikipedia

Es un algoritmo para medir la similitud entre dos secuencias temporales, que pueden variar en velocidad.
Se podrían detectar similitudes al caminar, incluso si una persona camina más rápido que otra, o si hubo aceleraciones y desaceleraciones durante una observación.

https://en.wikipedia.org/wiki/Dynamic_time_warping

Gemini

Es un algoritmo de aprendizaje no supervisado que se utiliza para comparar dos secuencias temporales. Permite que las secuencias de diferentes longitudes se alineen de manera óptima, incluso si las secuencias tienen diferentes velocidades o ritmos.

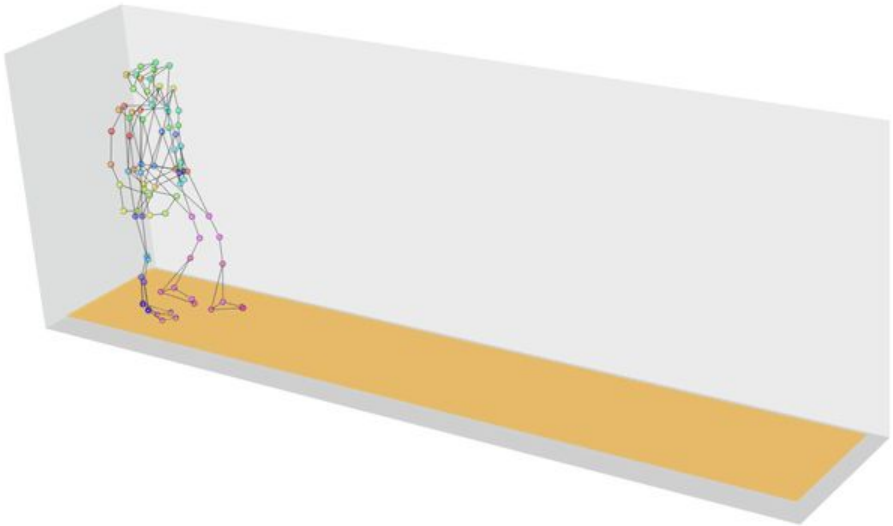
https://gemini.google.com/

¿Qué es DTW?

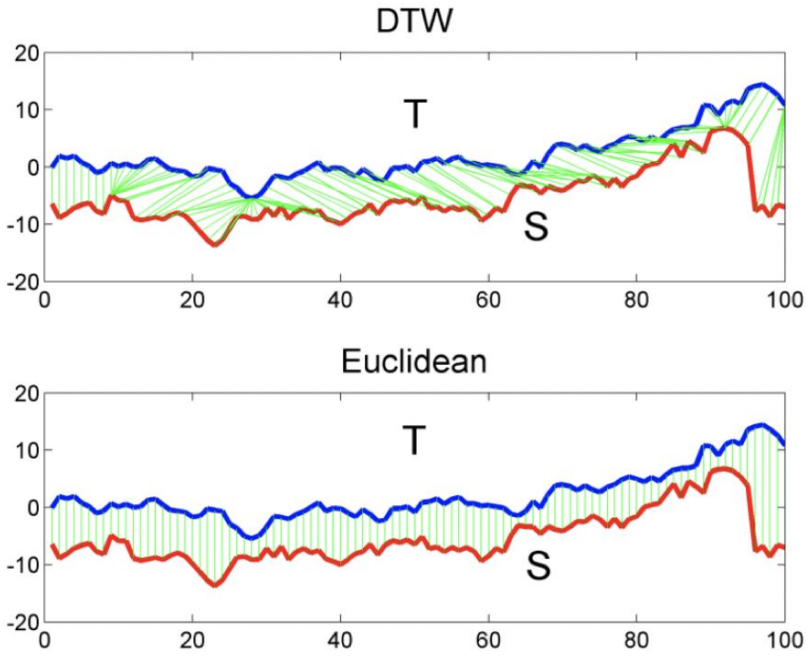
- El algoritmo de Deformación Temporal Dinámica (DTW) proporciona una mayor robustez al cálculo de **similitud** entre series temporales.
- Este método permite comparar series temporales de diferente longitud, ya que reemplaza la comparación punto a punto utilizada en la distancia euclidiana por una comparación **muchos-a-uno** (y viceversa).
- La principal característica de esta **medida de distancia** es que permite reconocer formas similares, incluso si presentan transformaciones de señal, como desplazamientos y/o escalamientos.

Ejemplo de DTW

Este algoritmo proporciona mediciones de distancia intuitivas entre señales al ignorar los cambios globales y locales en la dimensión del tiempo.



DTW vs Distancia Euclidiana



Matriz de distancia

- Dadas dos series de tiempo $T = \{t_1, t_2, \ldots, t_n\}$ y $S = \{s_1, s_2, \ldots, s_m\}$ de longitud n y m, respectivamente, un alineamiento del método DTW usa la información contenida en una matriz de distancia $n \times m$:

$$distMatrix = \begin{pmatrix} d(T_1,S_1) & d(T_1,S_2) & \cdots & d(T_1,S_m) \\ d(T_2,S_1) & d(T_2,S_2) & & \\ \vdots & & \ddots & \\ d(T_n,S_1) & & & d(T_n,S_m) \end{pmatrix}$$

- donde $distMatrix(i, j)$ corresponde a la distancia del i–ésimo punto de T y el j–ésimo punto de S, $d(T_i, S_j)$, con $1 \leq i \leq n$ y $1 \leq j \leq m$.

Warping path

- El objetivo de DTW es encontrar el camino de alineamiento (*warping path*) $W = \{w_1, w_2, \ldots, w_k, \ldots, w_K\}$ de elementos contiguos en $distMatrix$, de manera que minimice la siguiente función:

$$DTW(T,S) = \min\left(\sqrt{\sum_{k=1}^{K} w_k}\right)$$

Warping path

- El camino de alineamiento o deformación está sujeto a varias restricciones. Dados $w_k = (i, j)$ y $w_{k-1} = (i', j')$ con $i, i' \leq n$ y $j, j' \leq m$:
  1. **Condiciones de frontera**: $w_1 = (1, 1)$ y $w_K = (n, m)$. El camino debe comenzar en el primer punto de ambas series temporales y terminar en el último punto de ambas series temporales.
  2. **Continuidad**: $i\text{-}i' \leq 1$ y $j\text{-}j' \leq 1$. El camino debe avanzar de un punto a otro adyacente en ambas dimensiones. No puede saltar puntos.
  3. **Monotonía**: $i\text{-}i' \geq 0$ y $j\text{-}j' \geq 0$. El camino debe avanzar hacia adelante en el tiempo en ambas series temporales. No puede retroceder en el tiempo.

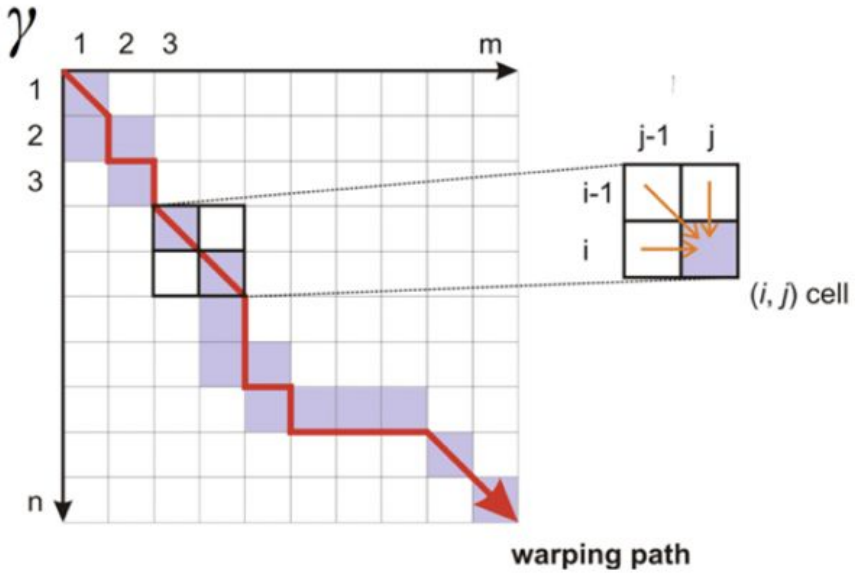
Warping path

- Se crea una **matriz de distancia acumulativa** γ de la misma dimensión que *distMatrix*, para almacenar en la celda (i, j) el siguiente valor:

$$\gamma(i,j) = d(T_i, S_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$$

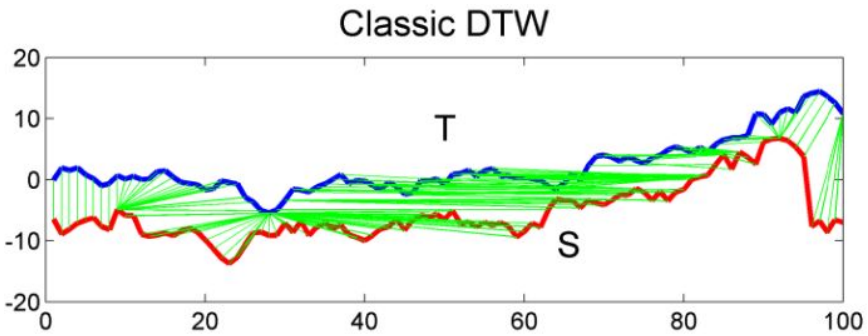
- El **último elemento** del *warping path*, w_K, corresponde a la **distancia** calculada con el método DTW.
- La complejidad general del método depende del cálculo de todas las distancias en la matriz *distMatrix*, que es $O(nm)$.

Warping path



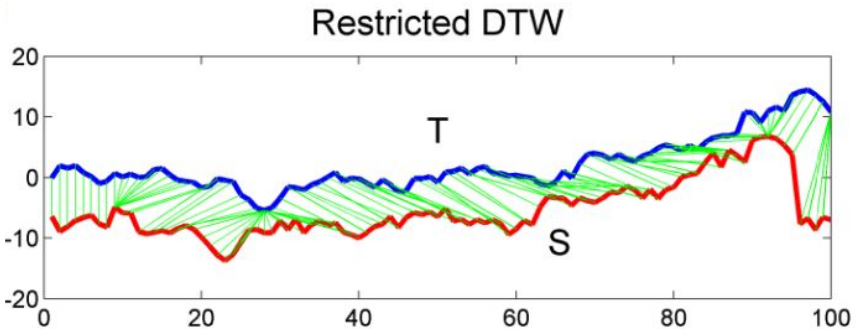
Restricción

- En muchos casos, este método puede llevar a efectos no deseados.
- Un ejemplo es cuando un gran número de puntos de una serie temporal T se asigna a un solo punto de otra serie temporal S.



Restricción

- Una forma de superar este problema es restringir el *warping path* de tal manera que tenga que seguir una dirección a lo largo de la diagonal.

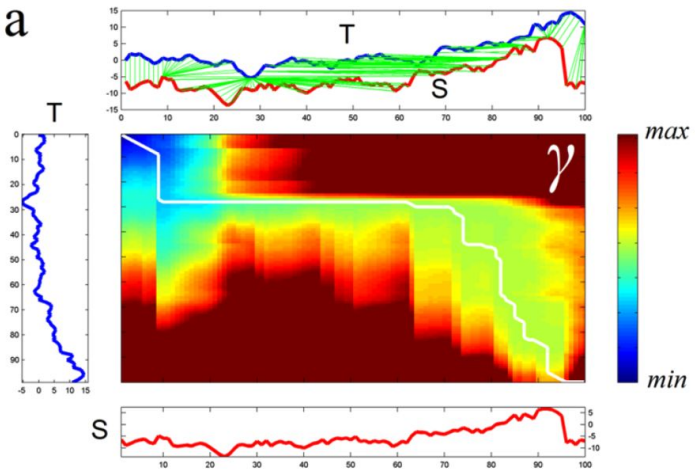


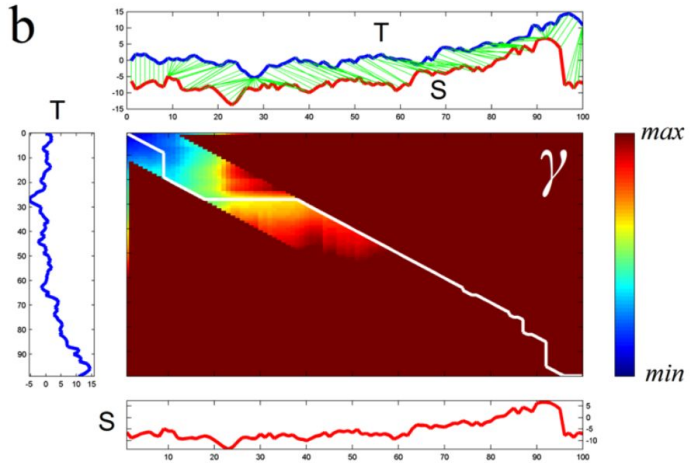
Restricción

- Para ello se restringe el camino imponiendo que la recursión se detenga a cierta profundidad, representada por un umbral δ.
- Entonces, la matriz de distancia acumulada *γ* se calcula como:

$$\gamma(i,j) = \begin{cases} d(T_i, S_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} & |i-j| < \delta \\ \infty & otherwise \end{cases}$$

Classic vs restricted implementation

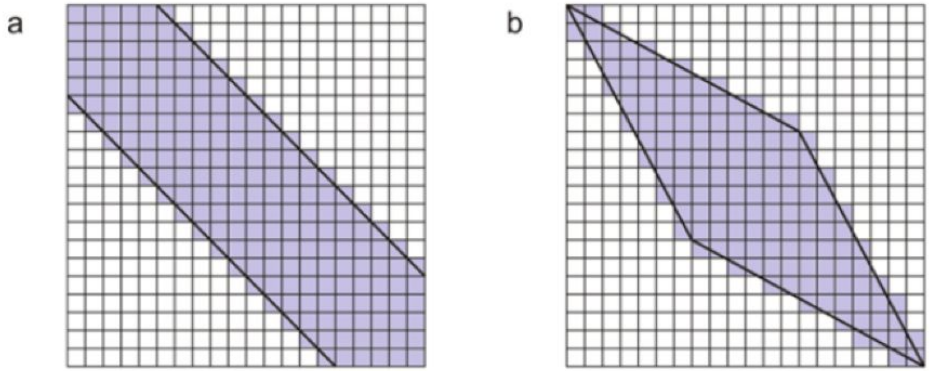




Restricción

- La restricción previa ($\delta = 10$), además de limitar asignaciones extremas, permite acelerar el cálculo de la distancia DTW.
  - Se almacenan distancias que estén a lo mucho δ posiciones alejadas (en dirección horizontal y vertical) de la diagonal de la matriz *distMatrix*.
  - Esto reduce la complejidad computacional a $O((n + m)\delta)$.
- Esta restricción se conoce como banda de Sakoe-Chiba, y se clasifica como una restricción global.
- Otra restricción global muy común es el paralelogramo de Itakura.

Restricciones globales



Algorithm 1 Cálculo de la Matriz de Distancia Acumulativa (DTW Clásico)

Require: Time series T of length n, time series S of length m
Ensure: Accumulated distance matrix dtw

Initialize dtw matrix of size $(n+1) \times (m+1)$ with ∞
Set $dtw[0][0] \leftarrow 0$
for $i \leftarrow 1$ to n **do**
  for $j \leftarrow 1$ to m **do**
    $cost \leftarrow |T[i-1] - S[j-1]|$
    $dtw[i][j] \leftarrow cost + \min(dtw[i-1][j],\ dtw[i][j-1],\ dtw[i-1][j-1])$
  end for
end for
return dtw

Algorithm 2 Estimación del Warping Path

Require: Accumulated distance matrix dtw of size $(n+1) \times (m+1)$
Ensure: Warping path $path$ and DTW distance $dtw[n][m]$

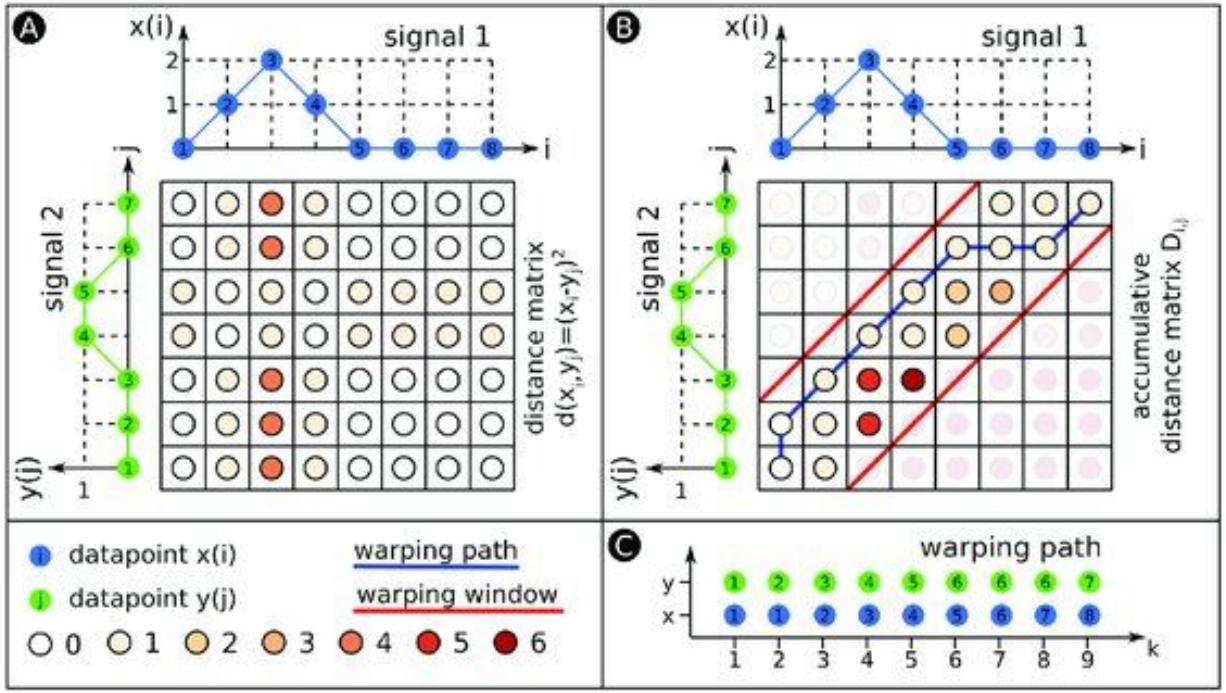
Initialize empty list $path$
Set $i \leftarrow n$, $j \leftarrow m$
while $i > 0$ **and** $j > 0$ **do**
  Append (i, j) to $path$
  if $dtw[i-1][j] = \min(dtw[i-1][j],\ dtw[i][j-1],\ dtw[i-1][j-1])$ **then**
    $i \leftarrow i - 1$
  else if $dtw[i][j-1] = \min(dtw[i-1][j],\ dtw[i][j-1],\ dtw[i-1][j-1])$ **then**
    $j \leftarrow j - 1$
  else
    $i \leftarrow i - 1$
    $j \leftarrow j - 1$
  end if
end while
while $i > 0$ **do**
  Append $(i, 0)$ to $path$
  $i \leftarrow i - 1$
end while
while $j > 0$ **do**
  Append $(0, j)$ to $path$
  $j \leftarrow j - 1$
end while
return $dtw[n][m]$, $path$ (reversed for correct order)

Algorithm 3 Cálculo de la Matriz de Distancia Acumulativa con Restricciones Globales

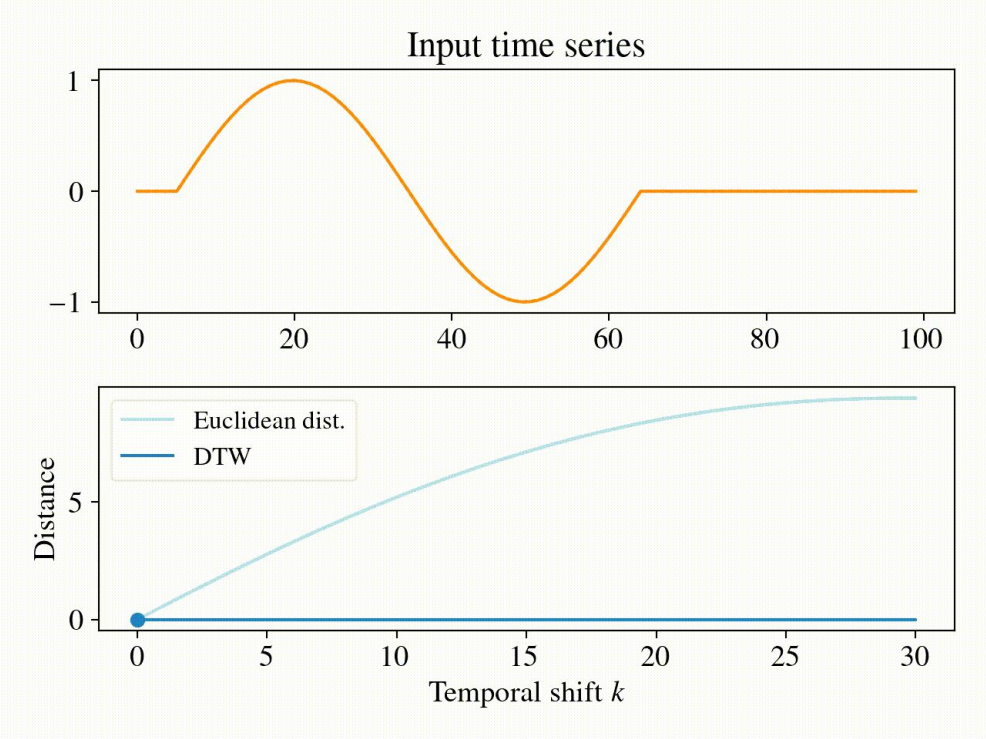
Require: Time series T of length n, time series S of length m, window size w for Sakoe-Chiba (optional)
Ensure: Accumulated distance matrix dtw with global constraints (Sakoe-Chiba or Itakura)

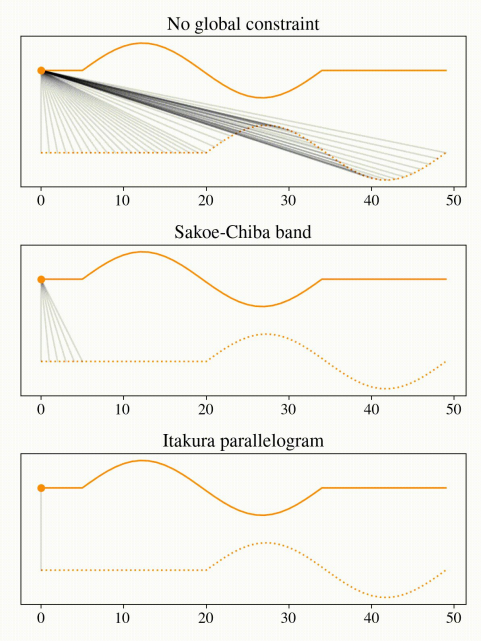
Initialize dtw matrix of size $(n+1) \times (m+1)$ with ∞
Set $dtw[0][0] \leftarrow 0$
for $i \leftarrow 1$ to n **do**
  for $j \leftarrow 1$ to m **do**
    if Sakoe-Chiba con ventana w **and** $|i-j| \leq w$ **then**
      $cost \leftarrow |T[i-1] - S[j-1]|$
      $dtw[i][j] \leftarrow cost + \min(dtw[i-1][j],\ dtw[i][j-1],\ dtw[i-1][j-1])$
    else if Itakura **and** $\frac{j}{m} - \frac{i}{n} \leq 1$ **and** $\frac{i}{n} - \frac{j}{m} \leq 1$ **then**
      $cost \leftarrow |T[i-1] - S[j-1]|$
      $dtw[i][j] \leftarrow cost + \min(dtw[i-1][j],\ dtw[i][j-1],\ dtw[i-1][j-1])$
    end if
  end for
end for
return dtw

Ejemplo de implementación



En resumen


Input time series
1
0
−1
0
20
40
60
80
100
Euclidean dist.
DTW
Distance
5
0
0
5
10
15
20
25
30
Temporal shift k

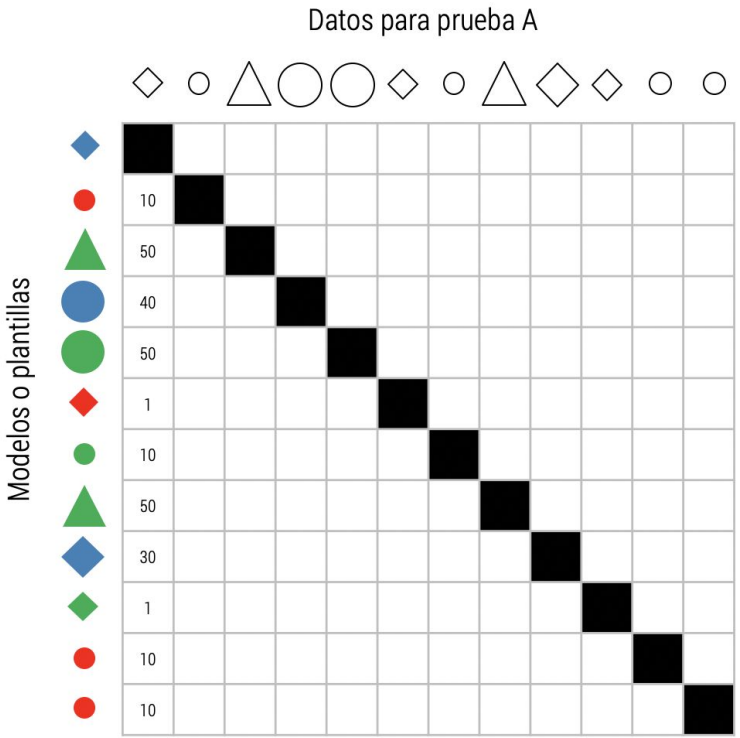

No global constraint
0
10
20
30
40
50
Sakoe-Chiba band
0
10
20
30
40
50
Itakura parallelogram
0
10
20
30
40
50

Algunas aplicaciones

- **Reconocimiento de voz**: se utiliza para comparar patrones de voz, en especial cuando las velocidades de habla varían entre diferentes hablantes.
- **Reconocimiento de gestos**: se puede aplicar para comparar y alinear gestos en sistemas de seguimiento de movimiento.
- **Reconocimiento de escritura a mano**: Puede utilizarse para comparar trazos de escritura a mano, especialmente cuando las velocidades de escritura varían entre diferentes individuos.

Clasificación

Esquema basado en plantillas (*soft*)

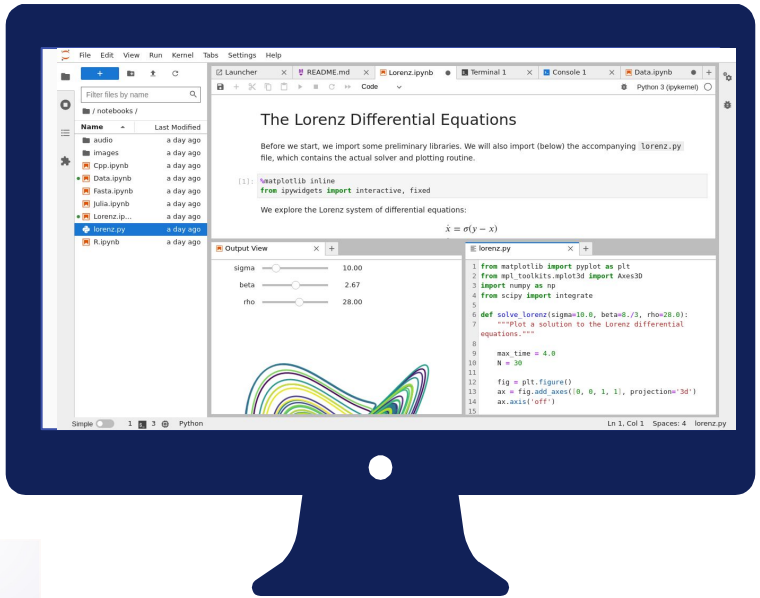


Para cada instancia en A, por ejemplo: ◇

Clase	Promedio de distancias	Clase más probable
rombo	(1+30+1)/3 = 10.6	rombo
círculo	(10+40+50+10+10+10)/6 = 21.6	
triángulo	(50+50)/2 = 50	

		Predicción		
		rombo	circulo	triangulo
Actual	rombo = 4	1	0	0
	círculo = 6	0	0	0
	triángulo = 2	0	0	0

(Go to live notebook)



Gracias!

¿Alguna pregunta?

hussein@cicese.mx
https://sites.google.com/view/husseinlopeznava

SCAN ME