
Distancia de Minkowski

Descripción de la métrica

La distancia de Minkowski es una métrica de distancia en un espacio vectorial normado que puede considerarse como una **generalización** de la distancia Euclídea y la distancia de Manhattan.

No es una distancia distinta por sí sola, sino una fórmula genérica que depende de un parámetro p . Al variar este valor, la métrica se transforma en otras distancias conocidas:

- Si $p = 1$, obtenemos la **Distancia de Manhattan**.
- Si $p = 2$, obtenemos la **Distancia Euclídea**.
- Si $p \rightarrow \infty$, obtenemos la **Distancia de Chebyshev**.

Fórmula Matemática

La fórmula para calcular la distancia de Minkowski entre dos puntos x e y en un espacio de n dimensiones es:

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

Qué tipo de datos compara

Se utiliza principalmente para comparar **datos numéricos** (vectores de números reales). Es ideal para:

- **Datos tabulares:** Comparar filas de un dataset donde las columnas son variables numéricas.
- **Algoritmos de Clustering y KNN:** Donde se necesita medir la similitud entre "vecinos".
- **Imágenes:** Si se representan como vectores de píxeles (aunque suele ser menos eficiente que otras métricas específicas para visión computacional).

Qué valores puede tomar

- **Rango:** $[0, \infty)$.
- Un valor de **0** indica que las dos instancias son idénticas.
- A medida que el valor aumenta, la "disimilitud" entre los puntos es mayor.
- **Requisito:** El parámetro p debe ser siempre ≥ 1 para cumplir con la desigualdad

triangular (una propiedad necesaria para ser considerada una "métrica" formal).

Implementación en Python

La librería más estándar y eficiente para esto es **SciPy**, específicamente dentro de su módulo de distancias espaciales.

- Enlace a la librería: [scipy.spatial.distance.minkowski](https://scipy.org/doc/stable/reference/generated/scipy.spatial.distance.minkowski.html)

Ejemplo de código Python

En este ejemplo, comparamos dos puntos en un espacio 3D usando un valor de $p=3$.

Python

```
from scipy.spatial import distance

# Definimos dos instancias (puntos en el espacio)
punto_a = [1, 2, 3]
punto_b = [4, 5, 6]

# Definimos el parámetro p
# p=1 es Manhattan, p=2 es Euclídea, p=3 es Minkowski de orden 3
p_valor = 3

# Calculamos la distancia
dist = distance.minkowski(punto_a, punto_b, p=p_valor)

print(f"La distancia de Minkowski (p={p_valor}) entre {punto_a} y {punto_b} es: {dist:.4f}")
```

Dynamic Time Warping (DTW)

Descripción de la métrica

Dynamic Time Warping es un algoritmo diseñado para medir la similitud entre dos secuencias que pueden variar en velocidad o ritmo. A diferencia de la distancia Euclídea, que compara los puntos de forma rígida (el punto i de una serie solo se compara con el punto i de la otra), DTW permite una **alineación elástica**.

Esto significa que puede "estirar" o "comprimir" secciones de una serie temporal para encontrar una correspondencia óptima con la otra. Es extremadamente útil cuando dos señales tienen la misma forma pero una está desplazada o dilatada en el tiempo.

Fórmula Matemática

DTW busca minimizar el costo acumulado de un camino de alineación en una matriz de distancias. Si tenemos dos series X e Y , se busca el camino W que minimice:

$$DTW(X, Y) = \min \sqrt{\sum_{k=1}^K w_k}$$

Donde w_k es el coste (distancia) de los puntos emparejados en el paso k , sujetos a restricciones de continuidad y monotonía.

Qué tipo de datos compara

Se utiliza casi exclusivamente para **datos numéricos secuenciales**:

- **Serie temporal:** Precios de acciones, telemetría de sensores, datos climáticos.
- **Audio:** Reconocimiento de voz (donde una persona puede hablar más lento que otra pero decir la misma palabra).
- **Biometría:** Reconocimiento de firmas o movimientos (gestos capturados con acelerómetros).

Qué valores puede tomar

- **Rango:** $[0, \infty)$.
- Un valor de **0** significa que las secuencias son idénticas (o pueden alinearse perfectamente sin coste).
- A mayor valor, mayor es el "esfuerzo" de estiramiento y la diferencia de magnitud necesaria para alinear las series.
- **Nota importante:** A diferencia de Minkowski, DTW no siempre cumple con la desigualdad triangular, por lo que técnicamente se le llama a veces "medida de similitud" más que métrica pura.

Implementación en Python

La librería de referencia para series temporales en Python es **DTAIDistance** (muy rápida por

estar escrita en C) o **FastDTW**.

- Enlace a la librería: [DTAIDistance en GitHub](#)

Ejemplo de código Python

Usaremos dtaidistance para comparar dos series temporales ligeramente desfasadas.

Python

```
from dtaidistance import dtw
import numpy as np

# Definimos dos series temporales (secuencias numéricas)
# La serie 's2' es similar a 's1' pero desplazada y con un valor extra
s1 = np.array([0, 0, 1, 2, 1, 0, 1, 0], dtype=np.double)
s2 = np.array([0, 1, 2, 1, 0, 1, 0, 0], dtype=np.double)

# Calculamos la distancia DTW
distancia = dtw.distance(s1, s2)

print(f"La distancia DTW entre las series es: {distancia:.4f}")

# Si calculáramos la Euclídea clásica, el valor sería mucho mayor
# porque los picos no coinciden en el mismo índice exacto.
```