**LIBRERÍA TRACLUS** (**[AdrielAmoguis](https://github.com/AdrielAmoguis/TRACLUS/commits?author=AdrielAmoguis)**)

1. **Variante del TRACLUS teórico que está usando**

Esta implementación sigue el marco general del algoritmo TRACLUS descrito por Lee, Han, y Whang (2007), centrado en la partición de trayectorias en segmentos y luego agrupando esos segmentos. Sin embargo, utiliza el algoritmo OPTICS para el agrupamiento en lugar de DBSCAN, que es el sugerido en algunas descripciones del TRACLUS (esto se puede cambiar en los parámetros de entrada).

1. **Longitud de cada cluster**

La longitud de cada cluster no se define explícitamente como un tamaño fijo en el código. En cambio, la longitud de un cluster se determina por la cantidad de segmentos de trayectorias que son agrupados juntos por el algoritmo OPTICS basado en su similitud. La similitud entre segmentos se calcula utilizando distancias como la euclidiana, perpendicular, paralela, y angular, y estas distancias se ponderan para obtener una métrica de distancia total que OPTICS utiliza para agrupar.

1. **Cómo segmenta las trayectorias**

El código segmenta las trayectorias utilizando la función **partition**, que divide una trayectoria en segmentos basándose en la descripción mínima longitud (**minimum\_description\_length**). Esta técnica evalúa si un segmento actual debe terminarse y comenzar uno nuevo basándose en si la adición de un nuevo punto aumenta la complejidad de la descripción de la trayectoria más allá de un cierto umbral. Esto se hace comparando el costo de descripción con y sin un nuevo punto de partición.

1. **Pasos de la ejecución del traclus**

**Validación de Entradas**

Primero, verifica que las trayectorias de entrada sean una lista de arrays de NumPy con la forma adecuada, asegurando que cada trayectoria esté correctamente formateada para el proceso de agrupamiento.

**Partición de Trayectorias**

Para cada trayectoria en el conjunto de datos, la divide en segmentos significativos utilizando la función **partition**. Esta función evalúa cada punto de la trayectoria para determinar puntos característicos que marcan cambios significativos en la dirección o en la velocidad, basándose en un cálculo de la descripción mínima longitud (**minimum\_description\_length**). El resultado es un conjunto de segmentos que representan partes de la trayectoria original. (cada punto que da el gps al principio)

**Conversión a Segmentos (si se requiere)**

Si se decide usar segmentos para el agrupamiento (use\_segments=True), convierte las trayectorias particionadas en una lista de segmentos individuales mediante la función **partition2segments**. Esto prepara los datos para el cálculo de distancias.

**Cálculo de la Matriz de Distancias**

Calcula la matriz de distancias entre todos los pares de segmentos utilizando la función **get\_distance\_matrix**. Esta matriz se basa en las distancias euclidiana, perpendicular, paralela y angular entre segmentos, ponderadas según los parámetros especificados. La matriz de distancias refleja la similitud entre todos los pares de segmentos en el conjunto de datos.

**Agrupamiento de Segmentos**

Utiliza el algoritmo OPTICS (se puede cambiar en los argumentos y usa **clustering\_algorithm**) para agrupar los segmentos basándose en la matriz de distancias. OPTICS identifica clusters basándose en la densidad de los segmentos, agrupando aquellos que son densamente conectados según los parámetros max\_eps y min\_samples.

**Asignación de Segmentos a Clusters**

Después de ejecutar OPTICS, asigna cada segmento a un cluster específico basado en las etiquetas de agrupamiento generadas. Los segmentos que no se asignan a ningún cluster se consideran ruido o puntos atípicos.

**Obtención de Trayectorias Representativas**

Para cada cluster, calcula una trayectoria representativa que resume los patrones de movimiento dentro del cluster. Esto se logra mediante la función **get\_representative\_trajectory,** que encuentra una línea o trayectoria que mejor representa la dirección y forma de los segmentos dentro de cada cluster.

**Retorno de Resultados**

Finalmente, la función retorna un conjunto de resultados que incluye las trayectorias particionadas, los segmentos, la matriz de distancias, los clusters con sus asignaciones correspondientes y las trayectorias representativas de cada cluster.