

Algoritmo de las Ventanas Deslizantes

función rolling()

Parámetros de entrada:

Data: Colección de objetos GeoData que contiene los datos georreferenciados.

Window_k: Número de datos de la ventana deslizante.

Window_x: Número de valores calculados sobre las ventanas deslizantes que se promedian.

Min_accuracy: Valor de precisión mínima para aceptar o rechazar un dato en la ventana deslizante.

Métrica: Métrica seleccionada (MEAN, MEDIAN).

Parámetros de salida:

Resultado: Promedio de **Métrica** sobre las **Window_x** ventanas deslizantes con la menor incertidumbre.

Incertidumbre: Promedio de la desviación estándar sobre las **Window_x** ventanas deslizantes con la menor incertidumbre.

Código_error: 1 OK, -1 hasta -4 Error.

Descripción del algoritmo:

1. Se realizan validaciones de los parámetros de entrada, si ocurre algún fallo se procede a almacenar 0 en <resultado> e <incertidumbre>, mientras que en **Código_error** se almacena:

- 1 si **Data** está vacía
- 2 si **Window_k** supera el número de elementos en **Data**
- 3 si **Min_accuracy** no está dentro del rango esperado [0, 1]
- 4 si no se pueden formar **Window_x** ventanas.

Y culmina el algoritmo. Caso contrario se continúa con el siguiente paso.

2. Se crea una cola FIFO para almacenar los datos (coordenadas GPS) de la ventana deslizante.

3. Se crea un Heap para almacenar los resultados de la métrica seleccionada y las desviaciones estándar por ventana, ordenándolos ascendentemente en función a la desviación estándar..

4. Se inicializa el contador del número de datos por ventana en cero.

5. Se recorren los elementos de la colección **Data**

5.1 Si el elemento actual tiene un accuracy mayor o igual al parámetro **Min_accuracy** entonces:

5.1.1 Se guarda el elemento actual en la cola

5.1.2 Se aumenta el contador del número de datos por ventana en 1.

5.1.3 Si el valor actual del contador de número de datos por ventana es mayor o igual al parámetro **Window_k** entonces:

5.1.3.1 Se calcula la métrica escogida con los datos de la cola.

5.1.3.2 Se calcula la desviación estándar de los datos de la ventana para estimar la incertidumbre de estos datos.

5.1.3.3 Se almacenan los resultados de la métrica y la desviación estándar en el Heap.

5.1.3.4 Se desencola un dato de la ventana y se encola el siguiente dato de la colección **<Data>**

6. Se desencola del Heap **Window_x** objetos.

7. Se calcula la media de las métricas y desviaciones estándares de los objetos seleccionados y se almacenan en **<resultado>** e **<incertidumbre>** respectivamente.

8. Se almacena 1 en **Código_error**

función rollingTemporal()

Parámetros de entrada:

Data: Colección de objetos TemporalGeoData que contiene los datos georreferenciados temporales.

Time_interval: Intervalo de tiempo que sirve como umbral para limitar la ventana deslizante.

Time_unit: Objeto ChronoUnit que indica la unidad de tiempo en la que se encuentra **Time_interval** (HOURS, MINUTES, SECONDS).

Window_x: Número de valores calculados sobre las ventanas deslizantes que se promedian.

Min_accuracy: Valor de precisión mínima para aceptar o rechazar un dato en la ventana deslizante.

Métrica: Métrica seleccionada (MEAN, MEDIAN).

Parámetros de salida:

Resultado: Promedio de **Métrica** sobre las **Window_x** ventanas deslizantes con la menor incertidumbre.

Incertidumbre: Promedio de la desviación estándar sobre las **Window_x** ventanas deslizantes con la menor incertidumbre.

Código_error: 1 OK, -1 hasta -6 Error.

Descripción del algoritmo:

1. Se realizan validaciones de los parámetros de entrada, si ocurre algún fallo se procede a almacenar 0 en <resultado> e <incertidumbre>, mientras que en **Código_error** se almacena:

- 1 si **Data** está vacía
- 2 si **Time_interval** es menor o igual a 0.
- 3 si **Min_accuracy** no está dentro del rango esperado [0, 1]
- 4 si **Time_unit** es una unidad de tiempo diferente a las especificadas.

Y culmina el algoritmo. Caso contrario se continúa con el siguiente paso.

2. Se crea una cola FIFO para almacenar los datos (coordenadas GPS) de la ventana deslizante.

3. Se crea un Heap para almacenar los resultados de la métrica seleccionada y las desviaciones estándar por ventana, ordenándolos ascendentemente en función a la desviación estándar..

4. Se inicializa el contador de ventanas y un contador de ventanas <vacías> en cero, donde las ventanas vacías representan a aquellas cuyos datos temporales no logran cubrir el umbral de tiempo requerido.

5. Se recorren los elementos de la colección **Data**

5.1 Si el elemento actual tiene un accuracy mayor o igual al parámetro **Min_accuracy** entonces:

5.1.1 Se guarda el elemento actual en la cola

5.1.2 Se consulta al primer elemento de la cola por su marca de tiempo y se calcula la diferencia temporal entre el primer elemento de la cola y el elemento actual.

5.1.3 Si la diferencia temporal cumple con el umbral establecido por **Time_interval** y **Time_unit** entonces:

5.1.3.1 Se calcula la métrica escogida con los datos de la cola.

5.1.3.2 Se calcula la desviación estándar de los datos de la ventana para estimar la incertidumbre de estos datos.

5.1.3.3 Se almacenan los resultados de la métrica y la desviación estándar en el Heap.

5.1.3.4 Se desencola un dato de la ventana y se encola el siguiente dato de la colección **<Data>**

5.1.3.5 Se aumenta en 1 el contador de ventanas.

5.1.4 Si la diferencia temporal no cumple con el umbral establecido por **Time_interval** y **Time_unit** y es mayor a este umbral entonces:

5.1.4.1 Se desencola un dato de la ventana.

5.1.4.2 Se aumenta en 1 el contador de ventanas <vacías>.

6. Si el contador de ventanas es menor a **Window_x** entonces se asigna 0 a los parámetros de salida <resultado> e <incertidumbre>, -5 a **Código_error** y se culmina el algoritmo. Caso contrario se continúa con el siguiente paso.

7. Se desencola del Heap **Window_x** objetos.

8. Se calcula la media de las métricas y desviaciones estándares de los objetos seleccionados y se almacenan en <resultado> e <incertidumbre> respectivamente.

9. Si el contador de ventanas vacías es mayor que 0 se almacena -6 en **Código_error**, caso contrario se almacena 1.