Prácticas de Algorítmica. 3º de Grado en Ingeniería Informática. Curso 2022-2023. Práctica 2.

Objetivos.

Con esta práctica se pretende que el alumno implemente dos algoritmos de segmentación de series temporales basados en la técnica divide y vencerás.

Definiciones.

Una serie temporal es una sucesión de observaciones de una variable realizadas a intervalos regulares de tiempo.



Figura: Serie temporal

Una segmentación de la serie temporal es un subconjunto de puntos (puntos dominantes o de corte) que conforman una representación más simple de la misma, conservando la información más relevante.

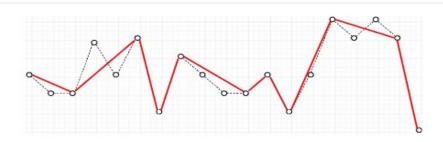


Figura: Segmentación de una serie temporal

Como se puede apreciar en la figura, la serie segmentada es una simplificación de la serie original, con lo que supone una compresión de la misma.

Evidentemente, hay diferencias entre la serie segmentada y la original. Existen dos formas de cuantificar esta diferencia de cara a ver la bondad de la segmentación:

- **Error máximo (eMax)**: Es la máxima distancia, medida en vertical, entre los puntos puntos de la serie y los segmentos rectos que componen la segmentación. En la figura anterior sería la distancia desde el cuarto punto al segmento que queda por debajo (aproximadamente 1.9).
- Suma de los errores al cuadrado (ISE): Es la suma de los cuadrados de las distancias en vertical de todos los puntos de la serie a los segmentos que la aproximan.

Enunciado:

Se han de implementar dos algoritmos de segmentación basados en el método divide y venceras, y usarán como medida del error el error máximo (eMax). El programa principal

contendrá un menú con dos opciones y se implementará de forma similar a como se hizo en la primera práctica, es decir, cada opción del menú llamará a una función de medio nivel sin parámetros y en esas funciones de medio nivel se implementarán cada uno de los métodos.

Para ello se suministra la clase SerieTemporal, y las clase Punto y Recta, que son clases auxiliares a la clase SerieTemporal.

La clase SerieTemporal contiene las siguientes funciones miembro:

- Constructor de la serie a partir de un fichero de puntos. Crea la serie, carga los puntos desde el fichero y los marca como no dominantes.
- Cosntructor de copia a partir de otra serie temporal.
- Observador y modificador para devolver o modificar el número de puntos de la serie.
- Observador y modificador para devolver o modificar un punto de la serie.
- Observador y modificador para ver o asignar un punto de la serie como dominante.
- Método para guardar los puntos de la serie temporal en un fichero.
- Método para guardar los puntos dominantes de la serie temporal en un fichero.
- Método para contar los puntos dominantes de la serie temporal.
- Método para calcular la suma de errores cuadráticos (ISE), el error máximo y su ubicación, de la segmentación realizada.
- Método para calcular la suma de errores cuadráticos entre dos puntos, suponiendo que estos están unidos mediante un segmento.
- Método para calcular el error máximo entre dos puntos y su ubicación, suponiendo que estos están unidos mediante un segmento.

Método para mostrar por pantalla los puntos dominantes obtenidos en la segmentación.

Algunos de estos métodos a su vez, hacen uso de los métodos de las clases Punto y Recta.

Los datos de entrada de ambos métodos serán el nombre del fichero que contiene la serie y el error máximo permitido.

Método 1. Segmentación a partir del punto del máximo error:

Se basa en obtener la segmentación buscando en cada momento el punto que genera el error máximo. Los pasos a seguir son los siguientes:

- 1. Marcar el primer y último punto de la serie como puntos dominantes. Este será el primer intervalo. (izquierda = 0, derecha = numeroPuntosSerie()-1)
- 2. Repetir de forma recursiva los siguientes pasos hasta que todos los intervalos entre puntos dominantes tengan un error inferior al error máximo.
 - 1. Calcular el error máximo y el punto correspondiente a ese error en el intervalo actual (posicionMaximo).
 - 2. Si el error es superior al eMax, marcar el punto de error máximo como dominante y aplicar recursivamente el método a los intervalos (izquierda, posicionMaximo) y (posicionMaximo, derecha). En caso contrario, ese intervalo ya forma parte de la solución.
- 3. Una vez segmentada la serie, guardar los puntos dominantes en un fichero y representar con gnuplot la serie y los puntos dominantes obtenidos en la segmentación.

Método 2. Segmentación a partir del punto del máximo error:

Se basa en obtener la segmentación buscando en cada momento el punto que genera el error máximo. Los pasos a seguir son los siguientes:

1. Marcar el primer y último punto de la serie como puntos dominantes. Este será el

- primer intervalo. (izquierda = 0, derecha = numeroPuntosSerie()-1)
- 2. Repetir de forma recursiva los siguientes pasos hasta que todos los intervalos entre puntos dominantes tengan un error inferior al error máximo.
 - 1. Calcular el error máximo correspondiente al punto intermedio entre izquierda y derecha.
 - 2. Si el error es superior al eMax, marcar el punto intermedio como dominante y aplicar recursivamente el método a los intervalos (izquierda, intermedio) y (intermedio, derecha). En caso contrario, ese intervalo ya forma parte de la solución.
- 3. Una vez segmentada la serie, guardar los puntos dominantes en un fichero y representar con gnuplot la serie y los puntos dominantes obtenidos en la segmentación.

Comprobación:

Se suministran varios archivos con series temporales, con extensión txt. Si se prueba el archivo BBVA.txt con un eMax=0.15 han de salir los siguientes resultados:

Método 1.

- número de puntos dominantes: 62
- *ISE* = 1156.22
- *errorMaximo= 1.4906*
- puntoErrorMaximo = 569

Método 2.

- número de puntos dominantes: 64
- ISE = 1073.9
- *errorMaximo=* 1.49206
- puntoErrorMaximo = 569

Otras series temporales suministradas:

Arrhytmia.txt, probad con eMax = 0.12 b41043.txt, probad con eMax = 1.8 HandOutlines, probad con eMax = 0.02 StarLightCurves.txt, probad con eMax = 0.04

Fecha de comienzo: 4 de octubre de 2022.

Fecha máxima de entrega: 18 de octubre de 2022.