Prácticas de Algorítmica.
3º de Grado en Ingeniería Informática.
Curso 2022-2023.
Práctica 4.

Objetivos.

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

Definiciones.

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

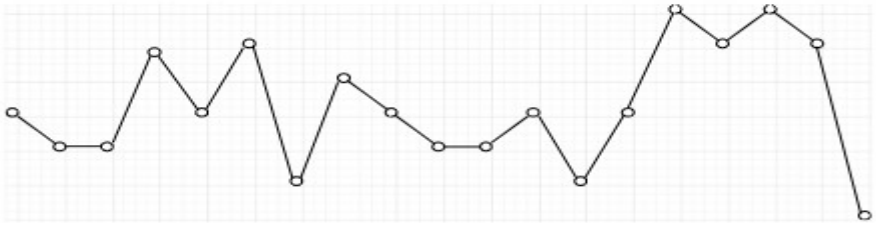

Figura: *Serie temporal*

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.

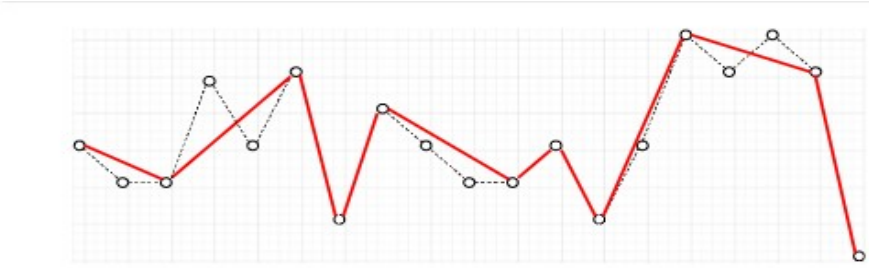

Figura: *Segmentación de una serie temporal*

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 programación dinámica. 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.*
- *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.* ***Esta función se ha cambiado con respecto a las prácticas anteriores, para poder acelerar el cálculo de los errores cuadráticos (ISE).***
- *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 número de puntos que contendrá la serie segmentada.
La salida del programa para ambos métodos será:

- *Valor de ISE de la solución óptima.*
- *eMax y punto con el que se corresponde.*
- *Tiempo que tarda.*

Método 1. Segmentación minimizando ISE, usando programación dinámica para calcular la solución óptima a todos los subproblemas posibles, hasta llegar la solución del problema inicial.
Para la implementación del método, usad como referencia el ejemplo desarrollado en el apartado 7.3.7 del tema 7, dedicado a la aproximación poligonal óptima.
Nota: Este método es obligatorio implementarlo. Nota máxima: 7.5

Método 2. Segmentación minimizando ISE, usando el algoritmo A, el cual solo calculará sólo la solución óptima de aquellos subproblemas que realmente conducen a la solución óptima del problema final.*
Para la implementación del método, usad como referencia el ejemplo desarrollado en el apartado 7.3.8 del tema 7, dedicado a la aproximación poligonal óptima basada en el algoritmo A.*
Para reducir el tiempo de ejecución, usad como valor de poda el doble del valor de ISE

obtenido por el método 1. La poda se aplicará en el momento de seleccionar los candidatos a formar parte de la lista de abiertos. Todo candidato cuyo error sea superior la valor de poda no llegará a entrar en la lista de abiertos.
Nota: Este método es opcional. Nota máxima: 2.5

<u>Comprobación:</u>
Se suministran varios archivos con series temporales, con extensión txt.
Dado que para las series que habéis usado anteriormente, los métodos de optimización pueden tardar mucho tiempo, las pruebas las debéis hacer con la serie BBVA50.txt, que posee solo 50 puntos. Una vez que funcione ya podéis probar la serie original del BBVA. Evidentemente, ambos métodos han de dar el mismo resultado.
BBVA50.txt con una segmentación de 5 puntos:

- *ISE = 7.27153*
- *errorMaximo= 1.01388*
- *puntoErrorMaximo = 11*
- *puntos óptimos de segmentación: 0, 5, 9, 10, 49.*

BBVA.txt con una segmentación de 62 puntos:

- *ISE = 387.193*
- *errorMaximo= 1.71817*
- *puntoErrorMaximo = 2562*

Fecha de comienzo: 8 de noviembre de 2022.
Fecha máxima de entrega: 22 de noviembre de 2022.