

Trabajo Práctico: Optimización

Toda la programación relativa a este trabajo debe ser realizada en Python (3.x). Si desea utilizar otro lenguaje o programa, por favor consultar con la cátedra.

Además de los archivos con los programas, se debe enviar un breve informe (en PDF) explicándolos. Al final de este informe, debe colocarse una copia de los programas para que sea más sencilla la anotación de comentarios sobre los mismos.

1. Problema: grupo par

1. Escriba una función que implemente la minimización basada en el método de gradientes conjugados usando interpolación cuadrática para la búsqueda lineal. La función debe tener el nombre `minimi` y recibir el handle a la función a minimizar, el handle a su gradiente, el punto inicial de la búsqueda (como arreglo de `Numpy`), la tolerancia de finalización y el número máximo de iteraciones, en ese orden. La tolerancia se debe interpretar así: si la norma de la diferencia entre dos puntos consecutivos es menor a la tolerancia, entonces el algoritmo debe terminar.
2. Escriba una función para que, sobre la base de `minimi` busque los parámetros a , b , c , T_1 y T_2 que minimicen

$$\frac{1}{N} \sum_i \left| y_i - \left(a + b \cdot \cos \left(2\pi \frac{t_i}{T_1} \right) + c \cdot \cos \left(2\pi \frac{t_i}{T_2} \right) \right) \right|^2,$$

donde t_i , y_i están en la primera y segunda columna del archivo `temp.txt`, respectivamente, y N es la cantidad de filas. Los valores de y_i se corresponden con la medición de temperatura corporal de una persona en cada hora.

El nombre de la función en este ítem debe ser `temperatura`, no debe recibir parámetros y debe devolver: un arreglo `Numpy` con los valores ajustados, un arreglo con el error del ajuste.

3. Escriba una función que pruebe el correcto funcionamiento de todas las demás funciones implementadas. La función debe llamarse `test`, sin argumentos.

Todas las funciones deben estar en un archivo denominado `temperamental.py`.

2. Problema: grupo impar

1. Escriba una función que implemente la minimización basada en el método de Nelder-Mead. La función debe tener el nombre `minimi` y recibir el handle a la función a minimizar, el handle a su gradiente, el punto inicial de la búsqueda (como arreglo de `Numpy`), la tolerancia de finalización y el número máximo de iteraciones, en ese orden. La tolerancia se debe interpretar así: si la norma de la diferencia entre dos puntos consecutivos es menor a la tolerancia, entonces el algoritmo debe terminar.
2. Escriba una función para que, sobre la base de `minimi` busque los parámetros a , b , c , T_1 y T_2 que minimicen

$$\frac{1}{N} \sum_i \left| y_i - \left(a + b \cdot \cos \left(2\pi \frac{t_i}{T_1} \right) + c \cdot \cos \left(2\pi \frac{t_i}{T_2} \right) \right) \right|^2,$$

donde t_i , y_i están en la primera y segunda columna del archivo `temp.txt`, respectivamente, y N es la cantidad de filas. Los valores de y_i se corresponden con la medición de temperatura corporal de una persona en cada hora.

El nombre de la función en este ítem debe ser `temperatura`, no debe recibir parámetros y debe devolver: un arreglo `Numpy` con los valores ajustados, un arreglo con el error del ajuste.

3. Escriba una función que pruebe el correcto funcionamiento de todas las demás funciones implementadas. La función debe llamarse `test`, sin argumentos.

Todas las funciones deben estar en un archivo denominado `temperamental.py`.