

Métodos Computacionales

Tarea 3 — 2018–20

Los archivos:

`Fourier.py`,

`Fourier2D.py`,

y `PCR.py`,

deben estar comprimidos en `ApellidoNombre_hw3.zip` y deben descomprimirse en un directorio `ApellidoNombre_hw3`. Este directorio comprimido debe contener únicamente los archivos mencionados anteriormente y debe subirlo a SICUA antes de las 8:00 pm del miércoles 30 de octubre de 2018 (10 puntos). Recuerde que es un trabajo completamente individual (no puede usar códigos que usted no haya escrito en su totalidad ni puede recibir ayuda de nadie cuando esté escribiendo su código). Recuerde también que los códigos deben correr sin botar errores en los computadores de computfísica (Y110B). Si su código no corre la nota de ese ejercicio será de cero. Para evitarlo comente la sección de código problemática. Recuerde que es buena práctica comentar el código y elegir buenos nombres para sus variables. Eso será tenido en cuenta durante el proceso de corrección de esta tarea. No se aceptarán trabajos entregados tarde ni por otro medio distinto a SICUA.

1. (10 points) **Github**

En esta tarea se evaluará el uso de Github de la siguiente manera:

- Los archivos de solución deben, además de subirse a SICUA, estar en un repositorio cuyo enlace debe ser enviado por SICUA cuando se envíe la solución de la tarea.
- Debe haber al menos 15 commits de la tarea en dónde se vea claramente la **evolución** del trabajo (no es válido hacer varios "commits" justo antes de la hora de entrega de la tarea).
- Los commits se deben hacer para los archivos individuales de la tarea a medida que su trabajo en cada archivo vaya avanzando.

2. (25 points) **PCA y PCR (principal component regression)**

La idea de este ejercicio es que exploren un método de regresión basado en el método de componentes principales. Este método, llamado principal component regression, puede ser un método muy útil para reducir la dimensionalidad de datos o para clasificar los datos. Este ejercicio se basa en datos actuales de pacientes con cáncer, para los cuales se tomaron más de 30 parámetros en las biopsias de manera automatizada. Adicionalmente, se tiene el diagnóstico de cáncer por un método manual. El objetivo del ejercicio es establecer si es posible usar PCA para automatizar el diagnóstico. Puede encontrar la descripción de los datos, así como los datos en: <http://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/cancer/WDBC/> Para este ejercicio debe escribir un script de python llamado `PCR.py` que:

- Almacene los datos del archivo `WDBC.dat`.
- Calcule, con su implementación propia, la matriz de covarianza de los datos y la imprima
- Calcule los autovalores y autovectores de la matriz de covarianza y los imprima (para esto puede usar los paquetes de `linalg` de `numpy`). Su mensaje debe indicar explícitamente cuál es cada autovector y su autovalor correspondiente.
- Imprima un mensaje que diga cuáles son los parámetros más importantes en base a las componentes de los autovectores

- Haga una proyección de sus datos en el sistema de coordenadas PC1, PC2 y grafique estos datos. Use un color distinto para el diagnóstico maligno y el benigno y la guarde dicha gráfica sin mostrarla en `ApellidoNombre_PCA.pdf`.
- Imprima un mensaje diciendo si el método de PCA es útil para hacer esta clasificación, si no sirve o si puede ayudar al diagnóstico para ciertos pacientes, argumentando claramente su posición.

3. (30 (+bono 3) points) **Transformada de Fourier: implementación propia, filtros e interpolación.**

En este ejercicio deben, usando su implementación propia de la transformada de fourier discreta, analizar y filtrar una señal. Deben además explorar qué ocurre si se hace una interpolación de los datos de la señal original y luego se hace análisis de Fourier. No olviden poner labels y leyendas en las gráficas. Para esto debe escribir un script de python llamado `Fourier.py` que:

- Almacene los datos de `signal.dat` y de `incompletos.dat`
- Haga una gráfica de los datos de `signal.dat` y guarde dicha gráfica **sin mostrarla** en `ApellidoNombre_signal.pdf`.
- Haga la transformada de Fourier de los datos de la señal usando su **implementación propia** de la transformada discreta de fourier.
- Haga una gráfica de la transformada de Fourier y guarde dicha gráfica **sin mostrarla** en `ApellidoNombre_TF.pdf`.
Esta gráfica debe ser en función de las frecuencias (bono de 3pts si no usa el paquete `fttfreq`. Indique esto con un mensaje en la terminal.)
- Imprima un mensaje dónde indique cuáles son las frecuencias principales de su señal.
- Haga un filtro pasa bajos con frecuencia de corte $f_c = 1000Hz$. realice la transformada inversa y haga una gráfica de la señal filtrada. Guarde dicha gráfica **sin mostrarla** en `ApellidoNombre_filtrada.pdf`.
- Escriba un mensaje en la terminal explicando por qué no puede hacer la transformada de Fourier de los datos de `incompletos.dat`
- Haga una interpolación cuadrática y una cúbica de sus datos `incompletos.dat` con 512 puntos. Haga la transformada de Fourier de cada una de las series de datos interpoladas.
- Haga una gráfica con tres subplots de las tres transformada de Fourier (datos de `signal.dat` y datos interpolados) y guárdela **sin mostrarla** en `ApellidoNombre_TF_interpol.pdf`.
- Imprima un mensaje donde describa las diferencias encontradas entre la transformada de Fourier de la señal original y las de las interpolaciones.
- Aplique el filtro pasabajos con una frecuencia de corte $f_c = 1000Hz$ y con una frecuencia de corte de $f_c = 500Hz$.
- Haga una gráfica con dos subplots (uno para cada filtro) de las 3 señales filtradas y guárdela sin mostrarla en `ApellidoNombre_2Filtros.pdf`.

4. (25 points) **Eliminación de ruido periódico de una imagen**

El propósito de este ejercicio es ver el potencial de la transformada de Fourier para filtrar imágenes, y en este caso particular, para eliminar el ruido periódico de una imagen.

Escriba un programa llamado `Fourier2D.py` que:

- Almacene la imagen `arbol.png` en una arreglo de numpy.
- Usando los paquetes de `scipy`, realice la transformada de Fourier de la imagen. Eligiendo una escala apropiada, haga una gráfica de dicha transformada y guárdela sin mostrarla en `ApellidoNombre_FT2D.pdf`.
- Haga un filtro que le permita eliminar el ruido periódico de la imagen. Para esto haga pruebas de cómo debe modificar la transformada de Fourier.
- Grafique la transformada de Fourier después del proceso de filtrado, esta vez en escala `LogNorm` y guarde dicha gráfica sin mostrarla en `ApellidoNombre_FT2D_filtrada.pdf`.
- Haga la transformada de Fourier inversa y grafique la imagen filtrada. Verifique que su filtro elimina el ruido periodico y guarde dicha imagen sin mostrarla en `ApellidoNombre_Imagen_filtrada.pdf`.