

Departamento de Matemática Aplicada
Universidad Rey Juan Carlos de Madrid
Ampliación de Matemáticas Aplicadas
Alexandru Iosif
(Curso 2022 - 2023)

TEMA 4 - LABORATORIO: PARTE 1 DE 2

Tiempo de exposición: 10 minutos

Porcentaje de la nota total de la práctica: 50%

Esta es la primera parte de la práctica. Guarde todas las funciones y todos los cálculos hechos. No va a tener que entregar un informe de esta parte (pero sí el código y los outputs en un documento .txt), sino que va a tener que exponer los cálculos (en vivo), en una fecha aún por determinar. Practique antes de la exposición y asegúrese de que las funciones ejecutan. Parte de la exposición podría ser contestar preguntas sobre el código. Durante la exposición tenga ya preparada la secuencia de funciones a ejecutar para ir lo más rápido posible.

1. En este ejercicio queremos crear una función en Matlab, Octave o Python (o sus derivados, como, por ejemplo, Sage) que haga una regresión lineal con el método de mínimos cuadrados. Va a tener que medir los tiempos de ejecución y compararlos entre sí. A continuación `**` se refiere a `.m` o a `.py`:

- Genere dieciseis puntos arbitrarios en el plano XY, dentro del cuadrado $[0, 10] \times [0, 10]$.
- Cree una función `regr.**` cuya entrada sean los primeros diez puntos y cuya salida sea una recta en el plano XY que represente la regresión lineal de estos datos con el método de mínimos cuadrados.
- A continuación lea los otros seis puntos y cree una función `regrRecurrente.**` que utilice el método usado en la clase de teoría para actualizar la regresión lineal de la siguiente manera:
 - Primero añadimos un punto y actualizamos la anterior regresión lineal de diez puntos.

- Luego añadimos los siguientes dos puntos y actualizamos la anterior regresión lineal de once puntos.
- Finalmente, añadimos los últimos tres puntos y actualizamos la anterior regresión lineal de trece puntos.
- En esta parte utilice el método usado en clase para actualizar la regresión lineal. No puede hacer la regresión de los once, trece y dieciseis puntos, respectivamente, desde cero, sino que tiene que usar el método de clase, aprovechando los cálculos de cada una de las regresiones anteriores.
- Cree otra función `regrRecurrenteCov.**` que añada cierta aleatoriedad al problema mediante una matriz de covarianza arbitraria (asegúrese de que es una verdadera matriz de covarianza) a las funciones anteriores. Use la parte de teoría sobre filtros de Kalman para ello.
- Represente cada una de las regresiones lineales.
- Si ha usado Matlab u Octave, utilice las funciones ya implementadas en Matlab u Octave para resolver estos mismos problemas y compare las soluciones de estas funciones con las soluciones de sus funciones así como los tiempos de ejecución.
- Comente las diferencias en los tiempos de ejecución.

Rubrica para la nota:

- Exposición (80%):
 - Las funciones ejecutan y los outputs son los que se piden: 40%
 - El alumnado comprende lo que ha hecho y es obvia la colaboración (20%).
 - El alumnado no ha necesitado más de diez minutos para la exposición (20%)
- Código (20%)
 - Estética y limpieza del código y el alumnado ha hecho un buen uso de las funciones ya implementadas en los distintos lenguajes de programación (por ejemplo, si se ha usado python, se han usado las funcionalidades para generar números aleatorios).

- Tened cuidado. Podéis usar código obtenido en Internet, pero citadlo. Un plagio significa una nota final 0 en esta práctica.
- También: mientras los grupos pueden colaborar entre sí para obtener ideas y despejar dudas, los códigos no pueden ser iguales, sino que cada grupo tiene que mostrar su código original. Si se detectan copia y pega de los códigos, la nota final de esta práctica será un 0. No podéis citar el código de un/a compañero/a.