Métodos Computacionales
Segundo Trabajo Práctico
Segundo Semestre 2023

Introducción

En el primer trabajo práctico se abordó el método de interpolación polinomial con el objetivo de aproximación de funciones. El objetivo de este segundo trabajo será estudiar un método alternativo de aproximación, el de *aproximación por cuadrados mínimos*.

Antes, conocíamos la expresión exacta de la función que buscabamos aproximar. Ahora, como sucede en muchos de los problemas de la industria, no conoceremos la expresión exacta de las funciones que buscaremos aproximar, sino que contaremos con *muestras* o *datos* de algunos de los valores que toman dichas funciones.

Importante: Si bien trabajaremos con datos, el objetivo no es tratar aspectos relacionados a la estadística o inferencia, sino que haremos lo que se denomina *ajuste de curvas* (curve fitting).

A partir de un conjunto de datos dado por n pares ordenados $\{(x_i, y_i)\}_{i=1}^{n}$ en $\mathbb{R}^2$, buscaremos construir una función de ajuste $f(x)$ que resulte en una buena aproximación de los valores de y. El enfoque de cuadrados mínimos nos permite ajustar funciones que sean combinaciones lineales de otras funciones, en cuyo caso hallar la función de ajuste se reduce a hallar los valores óptimos de los coeficientes que conforman la combinación lineal, también llamados *parámetros*.

Si quisiéramos esta vez *ajustar* un polinomio de grado $k \in \mathbb{N}$ lo que hacemos es proponer la forma funcional

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1} + a_k x^k$$

que se puede entender como una combinación de funciones del conjunto $B = \{1, x, x^2, x^3, ..., x^k\}$. Luego buscamos los parámetros $a_0, a_1, \ldots, a_k \in \mathbb{R}$ que hacen que $f(x_i) = y_i$ para todo $i = 1, 2, 3, \ldots, n$. Al igual que antes, esto resulta en buscar solución al sistema

$$\begin{aligned} a_0 + a_1 x_1 + \ldots + a_{k-1} x_1^{k-1} + a_k x_1^k &= y_1 \\ a_0 + a_1 x_2 + \ldots + a_{k-1} x_2^{k-1} + a_k x_2^k &= y_2 \\ &\vdots \\ a_0 + a_1 x_n + \ldots + a_{k-1} x_n^{k-1} + a_k x_n^k &= y_n \end{aligned}$$

a partir del cual construímos la matriz de diseño con los datos x_i:

$$X = \begin{bmatrix} 1 & x_1 & \ldots & x_1^{k-1} & x_1^k \\ 1 & x_2 & \ldots & x_2^{k-1} & x_2^k \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & \ldots & x_n^{k-1} & x_n^k \end{bmatrix} \in \mathbb{R}^{n \times (k+1)}$$

el vector de parámetros y el vector de los datos y_i:

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} \in \mathbb{R}^{k+1}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n.$$

Como ahora la matriz X podría ser rectangular, debemos resolver por cuadrados mínimos y obtenemos los parámetros óptimos $\hat{\mathbf{a}}$ tales que

$$\|\mathbf{y} - X\hat{\mathbf{a}}\| \leq \|\mathbf{y} - X\mathbf{a}\| \quad \text{para todo } \mathbf{a} \text{ en } \mathbb{R}^{k+1}.$$

recuperando así nuestra función de ajuste f.

Ejercicios

En cada uno de los ajustes solicitados por los siguientes ejercicios,

1. Formular el problema de cuadrados mínimos para el ajuste solicitado, escribiendo la matriz de diseño, el vector de parámetros, el vector de datos a ajustar, las ecuaciones normales, y las dimensiones de cada uno.
2. Hallar la solución al problema resolviendo las ecuaciones normales.
3. Calcular el error de aproximación dado por $\|\mathbf{y} - X\hat{\mathbf{a}}\|$.
4. Reportar los valores de los parámetros óptimos hallados redondeando a una cantidad de cifras significativas razonable.

Los archivos de datos pueden leerse usando la función np.load de la siguiente manera:

```
x, y = np.load("ejercicio_XX.npy")
```

Ejercicio 1. Ajuste lineal.

a) Hallar la recta que mejor aproxima el conjunto de datos `(1)` en el sentido de cuadrados mínimos.

b) Graficar en una misma figura los datos (x_i, y_i) junto con la recta ajustada.

c) ¿Qué valor se obtendría si se usara la recta ajustada para intentar estimar el valor de la función desconocida cuando $x = 5$?

Ejercicio 2. Ajustes polinomiales.

a) Hallar la parábola que mejor aproxima el conjunto de datos `(2)` en el sentido de cuadrados mínimos.

b) Repetir con una función cúbica.

c) Graficar en una misma figura los datos junto con ambas curvas halladas.

d) ¿Cuál es mejor?

Ejercicio 3. Ajustes no polinomiales.

El objetivo en esta sección es explorar otras formas funcionales no polinomiales. (Ver por ejemplo Series de Fourier). Tener en cuenta el *trade-off* entre cantidad de parámetros involucrados y el error de aproximación, es decir, estimar la mínima cantidad de parámetros que resulten en un ajuste razonable.

a) Para los conjuntos de datos `(3.i)` y `(3.ii)` por separado, proponer una forma funcional adecuada y hallar la curva de la forma propuesta que mejor aproxima los datos en el sentido de cuadrados mínimos.

b) Graficar en dos figuras distintas los conjuntos de datos y las curvas ajustadas en cada caso.

Ejercicio 4. Ajustes no lineales.

a) Hallar la curva de la forma $f(x) = a\,x^b$ que mejor aproxima el conjunto de datos `(4.a)` en el sentido de cuadrados mínimos.

Sugerencia: recordar qué propiedades cumplen los logaritmos y ajustar por cuadrados mínimos la función $\log(f(x))$.

b) Repetir con el conjunto de datos `(4.b)` y una curva de la forma $f(x) = a\,b^x$.

c) Graficar en dos figuras distintas los conjuntos de datos y las curvas ajustadas en cada caso.

Ejercicio 5. Cuadrados Mínimos Ponderados

En situaciones en las cuales los pares ordenados en el conjunto de datos no tienen la misma variabilidad o no pueden considerarse igualmente precisos, es deseable realizar un ajuste que considere estas diferencias. Por ejemplo, el uso de diferentes instrumentos de medición o diferentes condiciones de medición derivan en mediciones con distintas magnitudes para el error. Queremos realizar un ajuste por cuadrados mínimos

que considere la preferencia por datos más precisos. Para ello consideramos el método de cuadrados mínimos ponderados.

En este ejercicio los datos pueden leerse de la siguiente manera:

```
x, y, IW = np.load("ejercicio_XX.npy")
```

done IW son los pesos asignados durante la recolección de los datos.

a) Comentar brevemente el objetivo del método de cuadrados mínimos ponderados. Luego, deducir la solución de cuadrados mínimos ponderados.

b) Repetir los pasos del ejercicio 2, utilizando cuadrados mínimos ponderados con los datos de los archivos `ejercicio_5_i.npy` y `ejercicio_5_ii.npy`. Comparar con la solución de cuadrados mínimos convencional (sin considerar los pesos).

c) ¿Qué diferencias encuentra encuentra entre ambos conjuntos de datos? ¿Qué sucede con el ajuste de la función cuadrática al usar distintos pesos para los datos?¿y con el ajuste de una función cúbica?

d) Graficar en una misma figura el conjunto de datos y las curvas cuadráticas ajustadas. Mostrar en una figura lateral los pesos correspondientes a cada caso.

Condiciones de Entrega

La entrega del trabajo debe consistir en al menos dos archivos:

1. `codigo.ipynb:` un *jupyter notebook* que contenga todo el código utilizado, usando Markdown para describir las tareas realizadas y cómo fueron resueltas, fórmulas matemáticas, los resultados de sus experimentos y conclusiones, al estilo de un informe o reporte. Las únicas librerías de Python admitidas son `NumPy` y `Matplotlib`. El notebook debe ser entregado ya ejecutado, es decir, con las salidas de cada celda visibles. Además, el código debe ejecutar sin errores, es decir, debe ser reproducible sin la necesidad de cambios adicionales.
2. `informe.pdf:` el mismo notebook del ítem anterior pero exportado en formato pdf. Como antes, ya ejecutado para que las salidas de cada celda sean visibles en el pdf.

Además, se deben tener en cuenta las siguientes consideraciones:

- El trabajo debe realizarse en grupos de 2 alumnos.
- La fecha límite de entrega es el **domingo 26 de noviembre**.
- La entrega debe ser realizada a través del campus, y debe ser realizada por sólo 1 de los integrantes del grupo (no entregar por duplicado).
- En cada uno de los archivos entregados deben especificarse, arriba de todo, los nombres y apellidos completos de los integrantes del grupo.
- Las consignas sirven como guía de trabajo, se valorará el uso de técnicas vistas en clase como operaciones de vectores, matrices, y demás.
- Mantener el mayor nivel de prolijidad posible, tanto en los desarrollos como en el código. Las resoluciones deben estar escritas de forma clara y precisa.