Métricas, datos y calibración inteligente

Danna Sofía Hernandéz Cala*
Escuela de Física
Facultad de Ciencias

26 de septiembre del 2025

Índice

Resumen

En este reporte se trabajaron con datos tomados constantemente por un sensor de bajo costo en cierto periodo de tiempo, sobre la concentracion de PM2.5 en diferentes zonas y se compararon con datos referenciales. Haciendo uso de la distancia euclidea se pudo ver la diferencia entre ambos grupos de datos durante ciertos intervalos de tiempo, establecidos por valores arbitrarios de ventanas, donde una ventana de 180 minutos presentó los mejores resultados. Este se utilizó para realizar una relacion lineal entre los datos de referencia y los datos a calibrar, la cual permitio ver que tan valido es este modelo en base a una tolerancia de 5. Los resultados mostraron que este modelo es moderadamente efectivo incluso cuando se redujeron los datos a la mitad.

1. Introducción

En la actualidad es necesario la toma y control de datos a una gran escala para multiples necesidades, lo cual muestra la necesidad de manejar distintos sensores para obtener estos datos. Sin embargo, estos suelene ser de bajo costo y no presentan daños muy rigurosos. Es por esto que se busca un modelo efectivo para calibrar estos datos tomados por los sensores, en base a unos datos de referencia más exactos.

Para informe se utilizó como referencia los datos recolectados por diferentes estaciones de la AMB sobre las concentraciones de PM2.5 durante la franja de tiempo del 1 de octubre del 2018 hasta el

*e-mail: danna2240666@correo.uis.edu.co

31 de agosto del 2019. Además, se buscó calibrar un grupo de datos similar tomado desde el 3 de noviembre del 2018 hasta el 1 de septiembre del 2019. Para ello se utilizaron diferentes herramientas computacionales para ver la diferencia de los grupos de datos y la validez de la relacion lineal entre estos.

Los datos que se manejaron se encuentran en la siguiente carpeta de GitHub, donde el documento llamado 'Datos Estaciones AMB' contiene los datos de referencia y el documento 'mediciones clg normalsup pm25' contiene los datos a calibrar. Tambien incluye el codigo utilizado para la mayoria de calculos de esta asignacion. Carpeta de GitHub

2. Resultados y analisis

En esta asignación se utilizó principalmente herramientas de python para realizar los calculos necesarios y para trabajar con los datos que se proporcionaron. Las bibliotecas que se utilizaron fueron *pandas*, *NumPy* y *matplotlib.pyplot*.

Para empezar, es necesario conocer cual es la diferencia entre los datos de referencia y los datos del sensor a calibrar y, para ello, se utilizó la siguiente formula:

$$\mathcal{D}(\mathbb{D}_i, \hat{\mathbb{D}}_{\hat{i}}) = \sqrt{\sum_{i,\hat{i}}^{n,m} (\mathbb{D}_i - \hat{\mathbb{D}}_{\hat{i}})^2} \tag{1}$$

Donde $\mathbb{D}_i = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ es el conjunto de datos de referencia y $\hat{\mathbb{D}}_{\hat{i}} = \{(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), ..., (\hat{x}_m, \hat{y}_m)\}$ el conjunto de datos a calibrar, siendo las variables independientes el tiempo en que se tomó cada dato y la variable independiente la concentración de PM2.5. Como se puede notar, el tamaño de ambos conjuntos no es necesariamente el mismo y, además, los datos no se tomaron al mismo tiempo con los mismos intervalos. Para solucionar esto, se establecieron ventanas de tiempo arbitrarias que permitian clasificar los datos en ciertos intervalos de tiempo ξ_j, la cantidad de datos en cada intervalo dependeria del valor de dicha ventana. El codigo utilizado para esta parte se ve de la siguiente manera:

```python
def promedio_en_ventana(tiempos, valores, centro, w):
    mitad = pd.Timedelta(minutes=w/2)
    intervalo = tiempos.between(centro - mitad, centro + mitad)
    vals = valores[intervalo]
    return vals.mean() if not vals.empty else np.nan

def calcular_distancia(datos_ref, datos_iot, w):
    inicio = max(datos_ref["Date&Time"].min(),
                 datos_iot["fecha_hora_med"].min())
    fin = min(datos_ref["Date&Time"].max(), datos_iot["fecha_hora_med"].max())
    paso = pd.Timedelta(minutes=w/2)
    centros = pd.date_range(start=inicio+pd.Timedelta(minutes=w/2),
                            end=fin-pd.Timedelta(minutes=w/2), freq=paso)

    ref_vals, iot_vals = [], []
    for c in centros:
        pr = promedio_en_ventana(datos_ref["Date&Time"],
                                 datos_ref["pm25_ref"], c, w)
        pi = promedio_en_ventana(datos_iot["fecha_hora_med"],
                                 datos_iot["pm25_iot"], c, w)
        if not np.isnan(pr) and not np.isnan(pi):
            ref_vals.append(pr)
            iot_vals.append(pi)

    if len(ref_vals) == 0:
        return np.nan, []

    ref_vals, iot_vals = np.array(ref_vals), np.array(iot_vals)
    promedio = np.sqrt(np.sum((ref_vals - iot_vals)**2))
    return promedio, list(zip(ref_vals, iot_vals))
```

Figura 1: Codigo 1

Haciendo uso de este codigo es que pudimos determinar la distancia entre los datos de referencia con los datos a calibrar. Los valores fueron diferentes dependiendo de la hoja de datos que se escogio del documento.

	Acualago	Pilar	Giron	Normal	Caldas
Ventana [min]	Distancia				
15	560.14	452.82	537.08	454.71	496.71
30	560.19	453.01	537.33	454.82	496.78
60	768.72	619.14	734.44	625.01	680.31
120	519.24	412.29	489.81	412.37	453.15
180	404.43	311.58	374.19	307.82	346.92

Cuadro 1: Distancias para cada ventana

Al tener los resultados de la distancia, podemos observar que para cada caso, el mejor valor de ventana es de 180 minutos.

Teniendo en cuenta esta ventana, podemos comparar los valores de la variable dependiente en cada uno de los intervalos ξ_j y hacer ajuste de minimos cuadrados para encontrar la relacion lineal entre los datos de referencia y los datos a calibrar, la cual sería de la forma $y_j(\xi_j) = \alpha\hat{y}_i(\xi_j)$.

Además, para determinar el alcance validez de esta relacion lineal, se establecio una tolerancia o error permitido de 5 y se calculo cual seria el valor de referencia que se predice en base al valor de

α y los datos a calibrar $(\alpha\hat{y}_i(\xi_j))$, esto con el fin de calcular el error entre este valor de predicción y_{pred} y el valor de referencia y determinar cuantos de los datos cumplian con esa tolerancia, esto con ayuda de la siguiente parte de codigo:

```
y_pred = alpha * y_iot #f(xi)=af^(x^i)


tol_abs = 5.0  # µg/m³

errors = np.abs(y_ref - y_pred)
frac_within_abs = np.mean(errors <= tol_abs)*100
```

Figura 2: Tolerancia relacion lineal

Teniendo esto en cuenta, los valores de α junto con la cantidad de datos que cumple con la tolerancia, para cada caso, son:

	α	%
Acualago	0.5903	81,4
Pilar	0.8389	59,0
Giron	0.7654	58,7
Normal	0.7705	74,5
Caldas	0.6692	81,9

Cuadro 2: Caption

Donde los valores de α más cercanos a 1, el caso ideal donde los datos de referencia y los datos a calibrar son iguales, son aquellos cuya distancia entre los dos grupos de datos fue menor. Pero al mismo tiempo, estos mismos fueron los que contaron con menor porcentaje de datos que cumplen la tolerancia.

Por ultimo, se realizó un procedimiento similar pero dividiendo el numero de datos a su primera mitad. Con estos se calculó otra vez α y se utilizó para predecir la otra mitad de los datos calibrados, de los cuales tambien se calculo su diferencia con los datos de referencia para obtener la cantidad de datos que cumplian con la tolerancia, dando un resultado mayor al 70 % para cada uno de los casos.

3. Conclusiones

En conclusion, los datos presentaron una significativa dispersion respecto a los datos de referencia, variando qué tanto dependiendo de cada grupo de datos, pero siendo consistente el echo que

las ventanas de 180 minutos presentaron mejores resultados. A pesar de esto, se pudo lograr efectivamente calibrar los datos del sensor, ya que, incluso al reducir el numero de datos a la mitad, se logró que la mayoria de los datos cumplieran con una tolerancia de 5, permitiendo acercar el modelo considerablemente al caso ideal de $\alpha = 1$. Gracias a esto se tiene a la disposicion un metodo moderadamente efectivo y confiable, que nos permita mejorar la exactitud de los datos, sin embargo, aunque no es un proceso excesivamente complicado, no siempre sera la solucion mas efectiva por la gran cantidad de datos con los que se tiene que trabajar, aumentando la posibilidad de errores entre los datos de referencia y los datos del sensor.