

Parcial 1: Teoría de Aprendizaje de Máquina 2024-I

Profesor: Andrés Marino Álvarez Meza, Ph.D.
Departamento de Ingeniería Eléctrica, Electrónica, y Computación
Universidad Nacional de Colombia - sede Manizales

1. Instrucciones

- Para recibir crédito total por sus respuestas, estas deben estar claramente justificadas e ilustrar sus procedimientos y razonamientos (paso a paso) de forma concreta, clara y completa.
- La componente teórica de cada uno de los puntos deberá entregarse a mano. La componente práctica (programación), debe trabajarse sobre Colaboratory o Kaggle. Enviar link de GitHub con la solución teórica y de simulación al correo electrónico `amalvarezme@unal.edu.co` antes de las 23:59 del 31 de marzo del 2024.
- Los códigos deben estar debidamente comentados en las celdas de código, y discutidos/explicados en celdas de texto (markdown). Códigos no comentados ni discutidos, no serán contabilizados en la nota final.

2. Preguntas

- 2.1 (Valor 2.5 puntos). Sea el modelo de regresión $t_n = \phi(\mathbf{x}_n)\mathbf{w}^\top + \eta_n$, con $\{t_n \in \mathbb{R}, \mathbf{x}_n \in \mathbb{R}^P\}_{n=1}^N$, $\mathbf{w} \in \mathbb{R}^Q$, $\phi : \mathbb{R}^P \rightarrow \mathbb{R}^Q$, $Q \geq P$, y $\eta_n \sim \mathcal{N}(\eta_n|0, \sigma_\eta^2)$. Presente el problema de optimización (inferencia) y la solución del mismo para los modelos mínimos cuadrados, mínimos cuadrados regularizados, máxima verosimilitud, máximo a-posteriori, y Bayesiano con modelo lineal Gaussiano. Asuma datos i.i.d. Discuta las diferencias y similitudes entre los modelos estudiados.
- 2.2 (Valor 2.5 puntos) Genere una simulación sobre Python de los regresores por máxima verosimilitud y máximo a-posteriori, discutidos en el punto 2.1, para ajustar la señal: $t_n = \cos[x_n/3] + \cos[x_n/4] + \eta_n$, con $x_n \in [0, 24\pi]$, contaminada con ruido blanco Gaussiano η_n ($SNR_{dB} = 2[dB]$). Asuma mapeo $\phi(\cdot)$ del tipo polinomial de orden Q y prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0, \sigma_w^2)$. Simule 500 datos para entrenar los modelos y 200 para predecir. Incluya normalización por `MinMaxScaler()` de `sklearn` después de generar el mapeo no lineal.