

PyMC

Se quiere inferir la probabilidad p de que un estudiante haga trampa en un examen. Se propone el siguiente algoritmo de privacidad para entrevistar a los estudiantes luego del examen y preguntarle si hicieron trampa: Cada estudiante lanza una moneda (oculta al entrevistador). El estudiante responderá sinceramente si la moneda sale cara. Si sale ceca, el estudiante (en secreto) lanza una 2da moneda y responde “Sí, hice trampa” si el lanzamiento de la moneda sale cara, y “No, no hice trampa”, si sale ceca. De esta manera, el entrevistador no sabe si el resultado fue una declaración de culpabilidad, o una cara en un segundo lanzamiento de moneda y así se preserva la privacidad de los estudiantes. Si de 100 estudiantes entrevistados 30 dicen haber hecho trampa, y asumimos una distribución a priori uniforme para p :

(a) *Análisis teórico:*

- Si asumimos que el estudiante hizo trampa, calcular la probabilidad de que un estudiante diga que hizo trampa.
- Si asumimos que el estudiante no hizo trampa, calcular la probabilidad de que un estudiante diga que hizo trampa.
- Si asumimos un p conocido, calcular la probabilidad de que un estudiante diga que hizo trampa en función de p .
- Hallar la densidad a posteriori de p . Puede expresar el resultado utilizando constantes multiplicativas sin definir.
- Hallar una expresión para la probabilidad (predictiva) de que un nuevo estudiante diga haber hecho trampa. Expresar el resultado en función de la esperanza a posteriori.
- Hallar una expresión para la probabilidad (predictiva) de que un nuevo estudiante haga trampa, expresando el resultado en función de la esperanza a posteriori.

(b) *Generación de datos:* Crear el conjunto de entrenamiento descrito en el enunciado del problema. Reordenarlo al azar.

(c) *PyMC:*

- Construir el modelo bayesiano antes descrito en PyMC. El mismo debe contener la variable observable y la variable que representa si efectivamente el estudiante hizo trampa y respetar la causalidad del problema.
- Utilizando `model_to_graphviz` (PyMC), graficar la red bayesiana.
- Muestrear 3 cadenas del experimento. Elegir los parámetros *draws* y *tune* de manera que PyMC no genere advertencias (warnings) y de alcanzar un ESS (bulk) > 300 y un $\hat{R} \leq 1.01$ para p . : La función `summary` (arviz) puede ser útil.
- Utilizando `plot_posterior` (PyMC), graficar la densidad a posteriori.
- Muestrear la distribución a posteriori para aproximar la probabilidad predictiva de que un estudiante diga haber hecho trampa por cadena. : Para calcular probabilidades y esperanzas predictivas no es necesario (ni recomendable) muestrear la distribución predictiva.
- Muestrear la distribución a posteriori para aproximar la probabilidad predictiva de que un estudiante haga trampa por cadena.