Probabilidad y Estadística para Inteligencia Artificial
Ejercicios Clase 4

Lautaro Delgado

27 de julio de 2020

1. Ejercicio 2

Encontrar el estimador de máxima verosimilitud de la probabilidad de ceca p, dado que se tira una moneda 100 veces y salen 55 cecas.

En primer lugar, se reconoce que se trata de un problema con una distribución binomial, por lo que se planteará la expresión de verosimilitud correspondiente teniendo en cuenta los datos del experimento.

$$L(p|n=100, x=55) = \left(\frac{100!}{55!(100-55)!}\right) p^{55}(1-p)^{(100-55)} \tag{1}$$

El parámetro a variar en este caso es p, por lo tanto, se procede a derivar la expresión de máxima verosimilitud en función del mismo. Se toma inicialmente el logaritmo, ya que el valor que maximiza al logaritmo también maximiza la verosimilitud.

$$\ln L(p|n=100, x=55) = \ln\left(\frac{100!}{55!(100-55)!}\right) p^{55}(1-p)^{(100-55)} \tag{2}$$

$$\ln L(p|n=100, x=55) = \ln\left(\frac{100!}{55!(100-55)!}\right) + \ln p^{55} + \ln(1-p)^{(100-55)} \tag{3}$$

$$\ln L(p|n=100, x=55) = \ln\left(\frac{100!}{55!(100-55)!}\right) + 55 \cdot \ln p + (100-55) \cdot \ln(1-p) \tag{4}$$

$$\frac{\partial \ln L(p|n=100, x=55)}{\partial p} = 0 + 55 \cdot \left(\frac{1}{p}\right) - (100-55) \cdot \left(\frac{1}{1-p}\right) \tag{5}$$

Igualamos la derivada a 0 para obtener el valor de p que maximiza la verosimilitud:

$$\frac{\partial \ln L(p|n=100, x=55)}{\partial p} = 0 \tag{6}$$

$$55 \cdot \left(\frac{1}{p}\right) - (100-55) \cdot \left(\frac{1}{1-p}\right) = 0 \tag{7}$$

$$p = 0{,}55 \tag{8}$$

Se verifica el valor obtenido de p en la derivada segunda para asegurar que se trate de un máximo:

$$\frac{\partial^2 \ln L(p|n=100, x=55)}{\partial p^2} = -\frac{55}{p^2} - \frac{45}{(1-p)^2} \tag{9}$$

$$\left.\frac{\partial^2 \ln L(p|n=100, x=55)}{\partial p^2}\right|_{p=0{,}55} = -404{,}04 \tag{10}$$

Como el signo de la derivada segunda es negativo, se trata definitivamente de un máximo. El resultado obtenido no es más que el promedio, lo cual se verifica para cualquier caso de distribución binomial.

Para poder obtener mediante simulación el valor de p que maximiza la verosimilitud, se genera un vector de p con 1000 valores equiespaciados de 0 a 1. Mediante un bucle se calcula con la función de probabilidad de masa la verosimilitud y se obtiene su máximo.

A continuación se observan los gráficos obtenidos para la verosimilitud y su derivada respecto de distintos valores de p.

En la Figura 1, se puede observar el gráfico de la verosimilitud así como el valor de p que la maximiza.

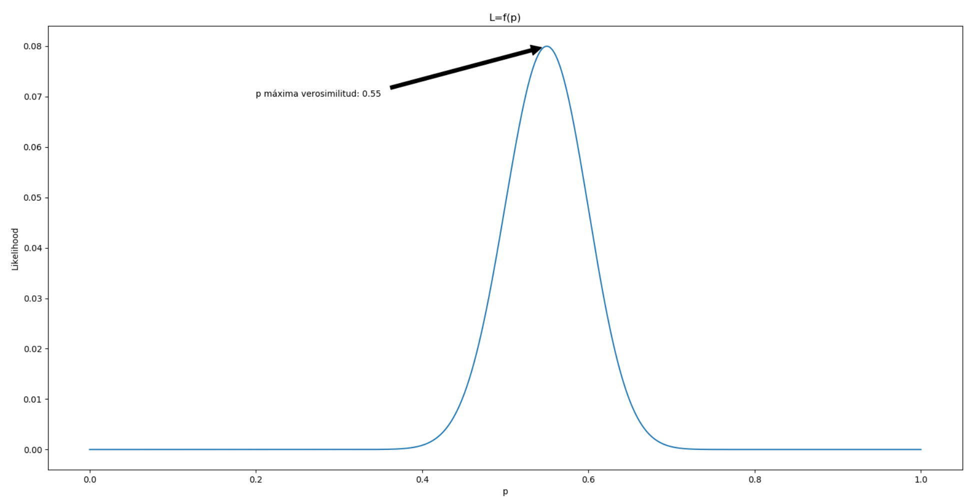

Figura 1: Verosimilitud en función de p

En la Figura 2, se muestra la derivada de la verosimilitud y cómo en el valor de p máximo, la misma es nula.

2. Ejercicio 3

Una v.a continua X responde a un proceso Gaussiano de media cero y $\sigma^2 = 1$. Simular varias realizaciones de X y estimar la pdf usando el método del histograma. Variar la cantidad de bins y sacar conclusiones acerca de la calidad de estimación.

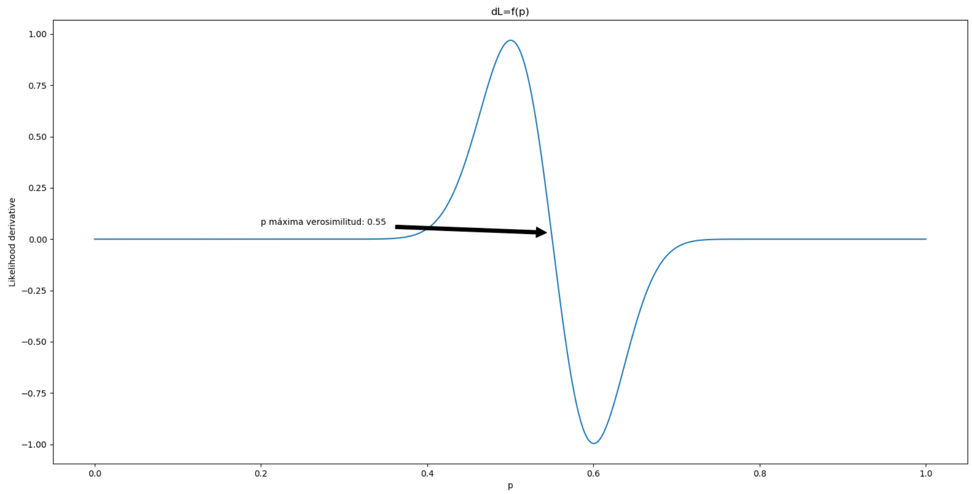

Figura 2: Derivada de verosimilitud en función de p

Se genera en Python una muestra de 1000 valores de X así como un vector de número de bins. Se realizan los histogramas para distintos valores de N = cantidad de bins y se calcula el Error Cuadrático Medio (MSE) respecto de la pdf real.

En la Figura 3, se muestran los histogramas armados de los valores de x muestreados para distintas cantidades de bins.

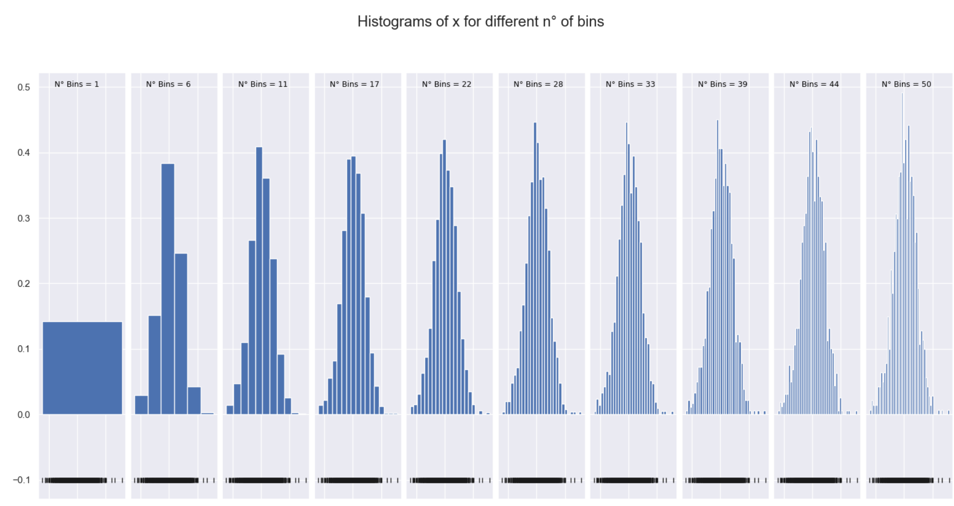

Figura 3: Histogramas para distintos valores de N

Para obtener el valor óptimo de cantidad de bins en relación al MSE, se realizan una serie de histogramas desde 0 hasta 1000 bins y se calcula en cada escenario el MSE respecto de la pdf real. En la Figura 4, se observar la evolución del MSE en función de la cantidad de bins. Se evidencia como el error decrece inicialmente hasta un mínimo (mejora el bias) pero luego vuelve a aumentar con N creciente (empeora la varianza)

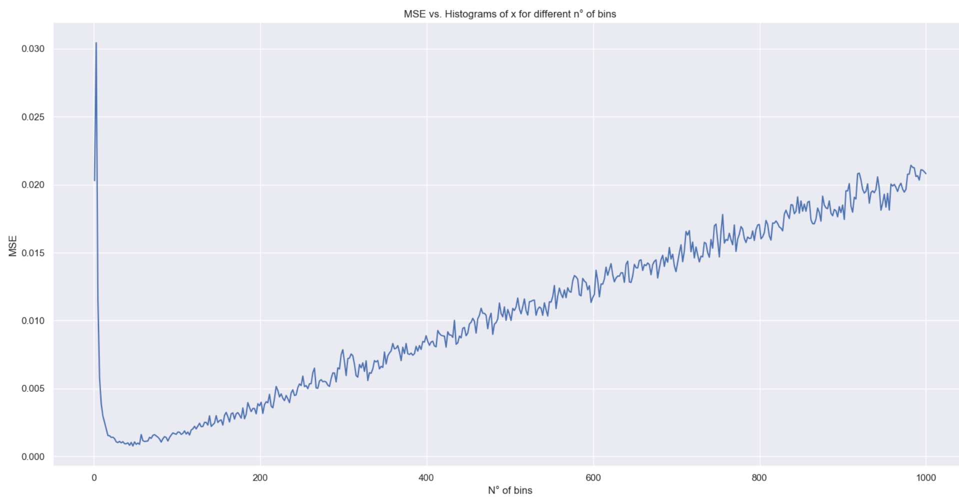

Figura 4: MSE en función de N

El escenario que minimiza el MSE es:

$$N_{optimo} = 47 \tag{11}$$
$$MSE = 7{,}66 \cdot 10^{-4} \tag{12}$$

En la Figura 5 se muestra el histograma suavizado resultante.

3. Ejercicio 4

En este ejercicio, se realiza el mismo procedimiento utilizando un estimador de kernel gaussiano. Nuevamente se implementa en Python una rutina que realiza la sumatoria del kernel aplicado a cada punto, calcula la densidad resultante y la normaliza (KDE). Posteriormente se comparan los resultados obtenidos con la rutina de KDE de Scikit-Learn.

En la Figura 6 se observa la estimación de pdf obtenida con la rutina de KDE implementada manualmente, utilizando un ancho de banda $h = 1$.

Por otro lado, en la Figura 7 se muestra la misma , para el mismo ancho de banda, obtenida mediante la librería Scikit-Learn.

En ambos casos el MSE obtenido es:

$$MSE_{h=1} = 2{,}47 \cdot 10^{-3} \tag{13}$$

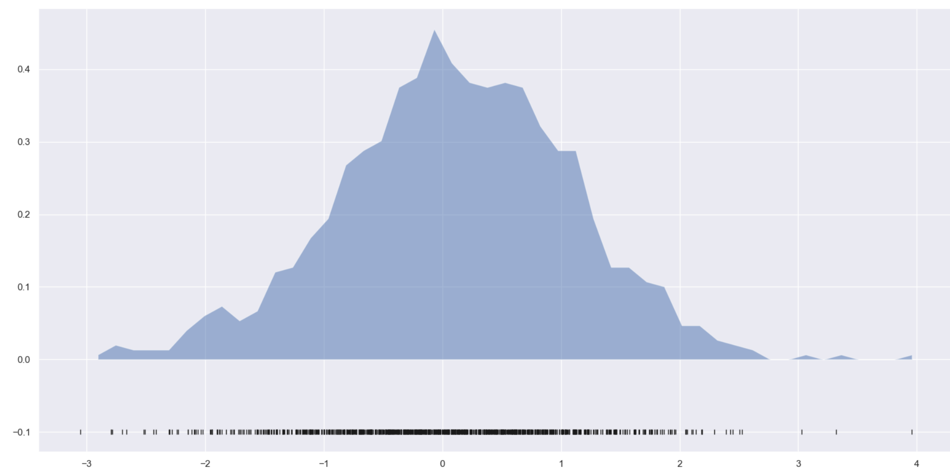

Figura 5: Histograma para N_{optimo}

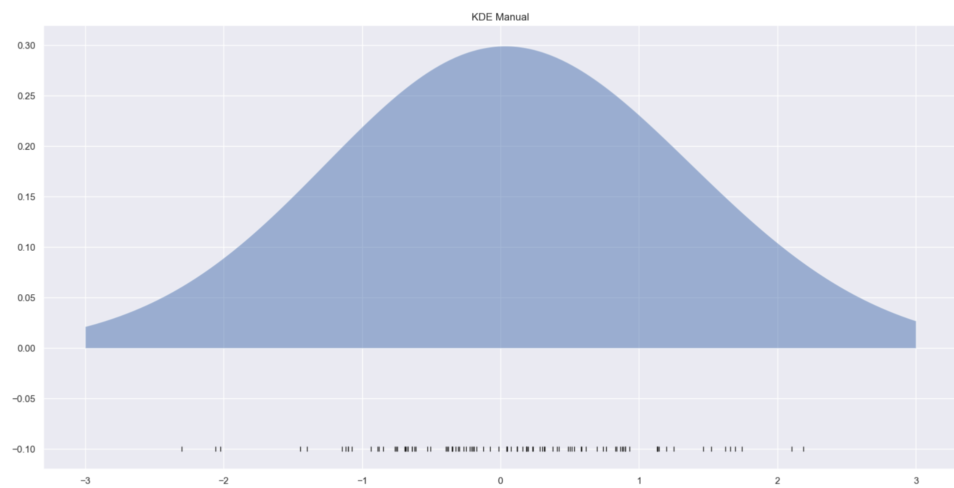

Figura 6: Gráfica de pdf obtenida mediante el KDE manual

De manera similar al ejercicio anterior, se calculan la evolución del MSE respecto de distintos anchos de banda para obtener el valor de h que minimiza el error. El comportamiento se puede observar en la Figura 8. En este caso para valores bajos de h el bias disminuye mientras que para valores altos de h, la varianza disminuye. Êl valor de h teórico óptimo es $h_{optimo} \propto n^{-\frac{1}{5}}$. En este caso los valores obtenidos son:

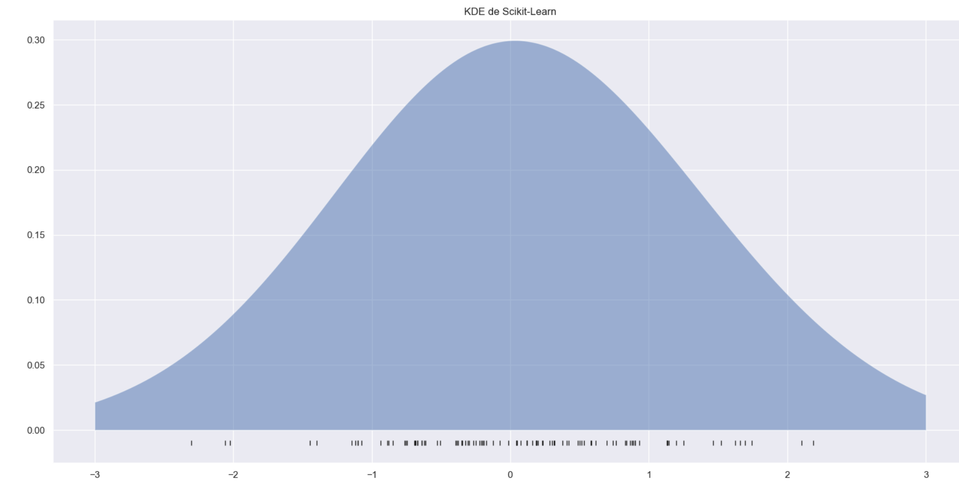

Figura 7: Gráfica de pdf obtenida mediante el KDE de SciKit-Learn

$$h_{optimo} = 0{,}5094 \tag{14}$$
$$MSE_{h_{opt}} = 1{,}05 \cdot 10^{-4} \tag{15}$$

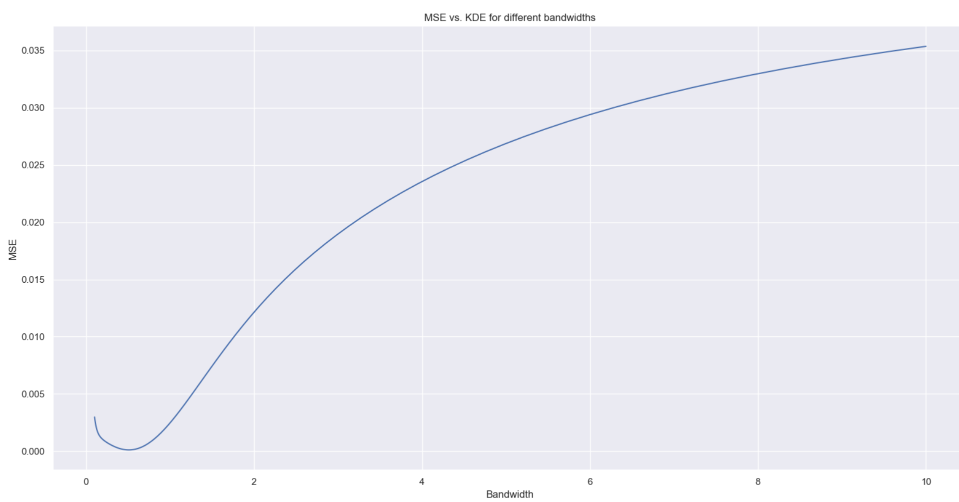

Figura 8: Evolución del MSE para KDEs con distintos valores de h

El ancho de banda óptimo también se puede obtener mediante una estrategia de Grid Search utilizando los métodos propios de la librería Scikit-Learn.