

Pruebas de uniformidad (y normalidad) multivariantes

Resumen

Se presenta una prueba de la hipótesis de que la distribución de una muestra de vectores aleatorios en \mathbf{R}^p es uniforme en el hipercubo $C = [0, 1]^p$ que puede interpretarse como una generalización de la prueba de Cramér - von Mises (C-vM) para el caso $p = 1$.

Si bien la hipótesis de uniformidad es extremadamente particular, la transformación integral de probabilidades (PIT) permite utilizar la prueba C-vM para probar la bondad de ajuste a cualquier distribución continua en \mathbf{R} .

De la misma manera, la prueba de uniformidad multivariante permite probar el ajuste a cualquier distribución continua en \mathbf{R}^p usando la Transformación de Rosenblatt, que generaliza la PIT, pero en forma mucho menos directa, trabajosa, y no canónica ya que hay $p!$ maneras diferentes de aplicarla.

Una extensión mucho menos general pero no carente de interés es la prueba de normalidad que se deriva de tipificar la muestra (aproximadamente, mediante la estimación de los parámetros) y aplicar la PIT a cada componente separadamente.

La prueba de uniformidad está basada en la convergencia del proceso empírico de la muestra a un proceso de Wiener típico W de parámetro en C condicionado por $W(1, 1, \dots, 1) = 0$ y en una descomposición de W en suma de 2^p procesos independientes cuyas normas en $L^2(C)$ tienen distribuciones de probabilidades que se describen y evalúan fácilmente.

Esa descomposición generaliza la del proceso de Wiener ordinario $w(t) = tw(1) + b(t)$ como suma de una *rampa* de pendiente aleatoria $w(1)$ más un *punte browniano* independiente b .

-0-

Se completa la presentación mostrando resultados de comparar mediante simulaciones el comportamiento de las pruebas de uniformidad propuestas con otras existentes en la bibliografía, especialmente la de Yang y Modarres implementada en el paquete SHT de R (The R Project for Statistical Computing) y de las de normalidad derivadas de las de uniformidad con las del paquete MVN de R.