UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS

MÉTODO DE INTEGRACIÓN TRADICIONAL DE MONTECARLO APLICADO A UNA DISTRIBUCIÓN EXPONENCIAL Y APLICACIONES DE BOOTSTRAP Y JACKKNIFE

Curso:

INTRODUCCIÓN A LA PROBABILIDAD Y ESTADÍSTICA

Integrantes:

BORJAS CORDOVA GUILLERMO JOEL
CALLUPE GANOZA FIDEL GABRIEL
ROJAS MINAYA NICOLAS GUILLERMO

Profesor:

LARA AVILA CESAR

LIMA - PERÚ
2018

Resumen

En este informe proponemos una alternativa particular para calcular la integral definida, emplearemos el lenguaje de programación R para la implementación y desarrollo del algoritmo mediante una función exponencial dada. Usaremos el método o simulación de Montecarlo para estimar el área bajo la curva en sus respectivos parámetros con lo que a continuación daremos a relucir la efectividad del método para muestras relativamente grandes. Finalmente pasaremos a dar un pequeña definición y aplicaciones que tienen los métodos de Bootstrap y Jackknife

Introducción

Método de Montecarlo:

Es un método numérico que permite resolver problemas matemáticos mediante la simulación de variables aleatorias. Lo vamos a considerar aquí desde un punto de vista didáctico para resolver un problema del que conocemos tanto su solución analítica como numérica.

Método de Bootstrap:

Es un método de remuestreo que se utiliza para aproximar la distribución en el muestreo de un estadístico. Su uso se centra generalmente en aproximar el sesgo o la varianza, así como construir intervalos de confianza o realizar contrastes de hipótesis sobre parámetros de interés

Método de Jackknife:

Es una técnica de muestreo especialmente útil para la varianza y el sesgo de estimación. El método Jackknife es anterior a otros métodos de remuestreo común, tales como el Bootstrapping. El estimador "Jackknife" de un parámetro se encuentra dejando sistemáticamente cada observación a partir de un conjunto de datos y el cálculo de la estimación, y luego encontrar el promedio de estos cálculos. Dada una muestra de tamaño N, la estimación Jackknife se encuentra mediante la agregación de las estimaciones de $N-1$ observaciones en la muestra.

Objetivo:

- Verificar que el método de integración de Montecarlo es válido para aproximar áreas.
- A partir del método de Bootstrap y Jackknife, obtener la media y la varianza de una muestra de datos, además de obtener intervalos de confianza.

Artículos Relevantes

a) Método de Montecarlo para calcular integrales: Muestra la aproximación (mediante un enfoque probabilista) de una integral definida de una función de una variable (que es continua y positiva) aplicando un método numérico tipo Montecarlo. El usuario puede elegir la función a integrar, el intervalo de integración, la tolerancia de la aproximación y la probabilidad. Se muestra el número total de puntos aleatorios que se necesitan generar para realizar el cálculo con la aproximación deseada así como el número de dichos puntos que caen dentro del recinto cuyo área determina la integral.

b) Aplicación del Método de Montecarlo para el cálculo de integrales definidas: En este trabajo se presenta un software educativo, desarrollado en matemática, para el cálculo de integrales definidas mediante el Método de Simulación o de Montecarlo. El programa realiza llamadas a un paquete en Mathematica que lleva a cabo la generación de los números aleatorios.

c) Métodos robustos de muestreos: Bootstrap y Jackknife
La aplicación de los potentes métodos estadísticos requiere de unas suposiciones, tales como la normalidad, que no siempre verifican los datos. Ésta es una de las razones de la presencia habitual de datos anómalos entre las observaciones muestrales. En estos casos es necesario emplear métodos estadísticos robustos.

d) Métodos cuasi Monte Carlo: Los métodos cuasi Monte Carlo constituyen una alternativa al método Monte Carlo tradicional para alcanzar resultados aproximados a problemas numéricos utilizando técnicas estadísticas. Este documento presenta conceptos básicos sobre los métodos cuasi Monte Carlo. Se analizan los principales aspectos teóricos involucrados en la definición del método, las formulaciones para el error cometido por el método y se presenta el análisis de dos trabajos que aplican técnicas estadísticas útiles en la práctica para obtener estimaciones del error.

Implementación en R

En el código R se observa dos métodos : fuerza "bruta" y Montecarlo por los cuales aproximar el área bajo la curva (exponencial):

Fuerza "Bruta":

En este método nosotros partimos análogamente al sucesor el cual primero definamos la curva de la función en este caso exponencial con sus respectivos paramentros pero a diferencia del Montecarlo verificaremos el área bajo la curva con un ciclo "for" el cual ira verificando uno a uno cada punto si pertenece o no a la gráfica,luego de comprobar la cantidad de puntos que corresponden respectivamente pasaremos a calcular la probabilidad de que un punto aleatorio "caiga" en nuestra área finalmente dicha probabilidad la multiplicaremos por el rectángulo encerrado para estimar una aproximación del área bajo la curva

Método de Montecarlo:

En el algoritmo partimos definiendo la curva (exponencial) usando la función "curve()" definiendo sus respectivos parámetros, establecemos el eje X de manera que vaya desde 0.5 hasta 3.5 y suprimiendo los ejes. Creamos una secuencia para definir las coordenadas X y Y;generamos un rectángulo dentro del cual se generaran las múltiples variables aleatorias, definimos el numero de muestras aleatorias(n) y hacemos uso de la función runif() el cual le dará aleatoriedad a las variables definimos un color para las variables que estén sobre (negro) y bajo(rojo) de la función. Mostaremos el valor real del área y el valor generado por la simulación.

Método de Bootstrap:

En el algoritmo definimos nuestra funcion "boot" que emulara el metodo de bootstrap, con sus respectivos argumentos "muestra"(es la muestra en la que trabajaremos), "rep"(determina el numero de repeticiones de la muestra original),"operacion"(determina el estadistico que queremos hallar),haremos uso de un buclo para poder realizar todas las repeticiones de las muestras, y acumular el estadistico que deseamos hallar,finalmente determinamos o calculamos la estimacion y error estandar de la muestra,para finalmente listar los resultados.

Método de Jackknife:

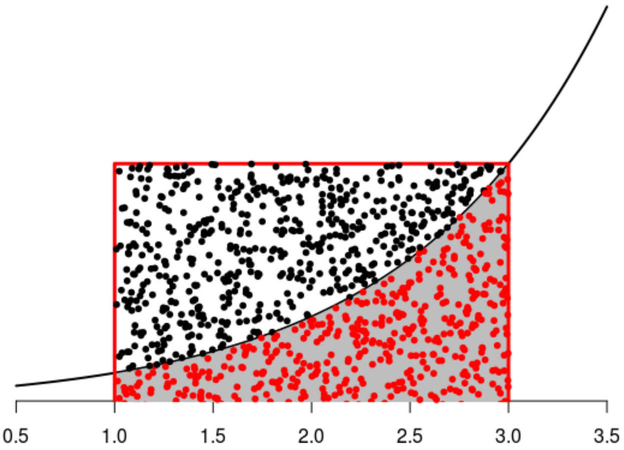
En el algoritmo definimos la funcion "jack" que simulara el metodo de jackknife, con sus variables "datos"(la muestra),"fun"(el estadistico que deseamos hallar), determinamos el tamaño de la muestra dada,y una variable "acumular[]"(que guardara nuestras estimaciones),haciendo uso de un bucle "for()",para almacenar el estadistico de interes, y calcular la estimacion, error estandar y sesgo, y mostrar los resultados,finalmente dada una muestra, haremos uso de diferentes tipos de estadisticos y hallar sus respectivos valores.

Experimentos y Resultados

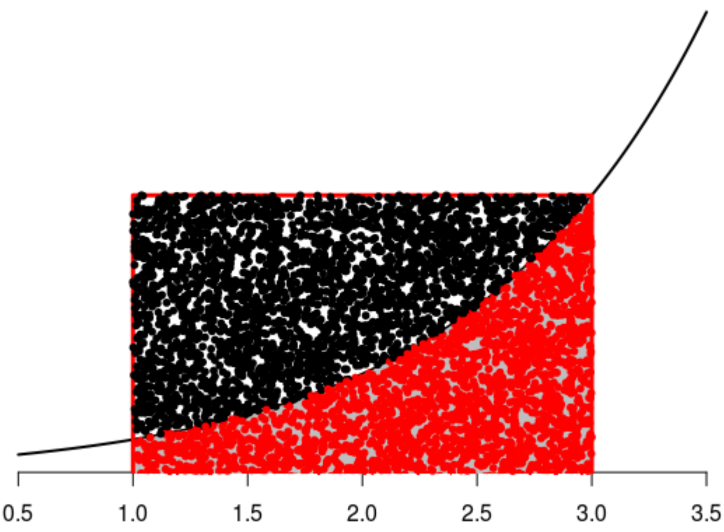
"Fuerza Bruta":

```
> #Establecemos los parametros
> t <-seq(1,3,by = 0.01)

> x <- c(1,t,3)

> y <- c(0,exp(t),0)

> #Creamos un contador en 0 quien llegara a ser la cantidad de puntos aleatorios
> contador <- 0

> #Definimos el numero de puntos aleatorios
> n = 1000

> #Definimos un bucle for que ira distinguiendo el numero de puntos bajo la grafica
> for (i in 1:n) {
+   x <-runif(n,1,3)
+   y <-runif(n,exp(1),exp .... [TRUNCATED]

> #Calculamos la probabilidad
> prob <-(contador/n)

> prob
[1] 0.321

> #Calculapmos el area bajo la curva aprox
> area <- prob*2*exp(3)

> area
[1] 12.89491

> #Comparamos con la verdadera area
> exp(3)-exp(1)
[1] 17.36726
```

"Metodo de Monte Carlo":




```
> #valor real del area
> exp(3)-exp(1)
[1] 17.36726

> #hallamos el area aproximada
> table(ab)/n
ab
    0     1
0.555 0.445

> (table(ab)/n)*2*exp(3)
ab
       0        1
22.29495 17.87613
```



"Bootstrap":

```
> #el primer resultao es la media bootstrap
> #el segundo es el error estandar del estimador
>
> #ahora probemos con otros estadísticos
>
> #para la .... [TRUNCATED]
$estad.b
[1] 9.732295

$ee.b
[1] 0.6809909


> #para el percentil 30
> boot(mues,200,quantile,0.3)
$estad.b
[1] 8.396447

$ee.b
[1] 0.4803702


> #para el cuantil 0.987
> boot(mues,200,quantile,0.987)
$estad.b
[1] 14.6572

$ee.b
[1] 0.9914773
```

"Jackknife":

```
> #jackknife
> #llamamos a nuestra funcion jack
> jack=function(datos,fun,...){
+   #variables: datos que sera la muestra a trabajar
+   #fun: el esta .... [TRUNCATED]

> muestra=rexp(50,1/2)

> #para la diferentes tipos de estadisticos
> jack(muestra,mean)
     est.clas    est.j      ee.j sesgo
[1,] 1.715107 1.715107 0.2760915     0

> jack(muestra,median)
     est.clas    est.j      ee.j sesgo
[1,] 1.232573 1.232573 0.5074288     0

> jack(muestra,quantile,0.25)
      est.clas     est.j       ee.j      sesgo
[1,] 0.3641955 0.3644491 0.07787343 0.01242753

> jack(muestra,IQR)
     est.clas    est.j      ee.j       sesgo
[1,] 1.740886 1.739639 0.2871011 -0.06110937
```

Discusión

- Para cada compilación del algoritmo obtuvimos resultados cercanos al valor del área real
- Cada vez que aumentábamos "n" la aproximación del área era mas exacta
- Es muy notoria la diferencia que hay entre el método de calcular el área por fuerza bruta que por el método de Monte Carlo se observa que, Monte Carlo es mucho mas eficiente ya que para una cantidad mucho menor de puntos aleatorios se obtiene una gran aproximación.
- El Método de Monte Carlo es un método de simulación que en cierta parte puede llegar a tener una precisión optima, sin embargo requiere de un gran consumo de recursos computacionales.
- El método de Bootstrap y Jackknife son muy eficientes en cuanto al calculo general del sesgo o varianza ya que se aproximan mucho a los valores reales.

Conclusiones

- Logramos comprobar que el método de integración de Montecarlo, nos dio una aproximación al área original bajo la curva exponencial.
- También pudimos comprobar que si generamos mas muestras aleatorias (mientras n sea mas grande), la aproximación que generaba la simulación de Montecarlo era mas cercana al resultado real lo cual nos permite afirmar que a mas muestras aleatorias, el área resultante de la simulación es mas exacta.
- El método de Bootstrap y Jackknife si bien tiene buen desempeño en sus tareas requieren de muchos recursos, en la actualidad existen variantes mas eficientes como $VarianteBWR$.
- El Método de Monte Carlo ,Bootstrap y Jackknife rinden papeles importantes sin embargo para cálculos de gran envergadura existen métodos mucho mas sofisticados para su desarrollo.

Verificación

- Metodo de fuerza bruta: `https://bit.ly/2ywY4tr`
- Metodo de Monte Carlo: `https://bit.ly/2KdtoPl`
- Bootstrap: `https://bit.ly/2luSnTv`
- Jackknife: `https://bit.ly/2yCi4Lc`

Bibliografía

- M. H. Kalos, P. A. Whitlock,Monte Carlo Methods,Wiley, 2nd edition, 2008.
- Peña Sánchez de Rivera, Daniel (2001). "Deducción de distribuciones: el método de Monte Carlo", Fundamentos de Estadística. Madrid: Alianza Editorial. ISBN 84-206-8696-4.
- R. Y. Rubinstein, D. P. Kroese, Simulations and the Monte Carlo Method, Wiley, 2nd edition, 2008.
- Third Edition (2005) Fundamental of probability with Stochastic Procceses Saeed Ghahramani Western New England College cap 13 pag 626
- `https://bit.ly/2K17DGh`
- `https://bit.ly/2KerNJ1`
- `https://bit.ly/2MPC7c9`