Approximation de MacLaurin et fonction de répartition

Julien Parfait Bidias Assala

2024-08-24

Problème

Considérons la densité de probabilité de la variable aléatoire X définie $\forall k \in \mathbb{R}^+$ par

$$f(x) = \rho e^{k^2}$$

Où $\rho > 0$ est un paramètre d'échelle. L'objectif est de calculer la fonction de répartition de f notée F. En effet,

$$F_X(k) = \int_0^k f(x)dx = \int_0^k \rho e^{x^2}\,dx.$$

Fonction de répartition

Après avoir effectué un DL de MacLaurin au voisinage de 0 de la fonction e^{x^2} et quelques manipulations cosmiques j'obtiens finalement :

$$\int_0^k e^{x^2}\,dx = \rho \sum_{n=0}^{\infty} \frac{k^{2n+1}}{(2n+1)\cdot n!}.$$

Ce résultat est-il correct ? C'est là où réside tout le problème quand on a pas les outils pour le vérifier. Pour vérifier ce résultat on peut comparer le calcul direct de l'intégrale avec un ordinateur ou une calculatrice au calcul avec un algorithme prenant en compte l'expression trouvée. Supposons que $\rho = 1$ et $k = 1$. Let's go !

La preuve

Dans R pour calculer une intégrale on peut utiliser la fonction **integrate** qui prend pour paramètres votre fonction, la borne inférieure et la borne supérieure comme on peut le voir à travers le programme ci-dessous : on obtient un résultat avec une marge d'erreur.

```r
# on cree la fonction f
f  <- function(x){exp(x^2)}
# borne superieure
k <- 1
# On calcule ensuite son integrale
resultat.1 = integrate(f, 0, k)
# On obtient le resultat suivant
resultat.1
```

```
1.462652 with absolute error < 1.6e-14
```

Vérification avec notre fonction

Rappel : $\int_0^k e^{x^2}\,dx = \rho \sum_{n=0}^{\infty} \frac{k^{2n+1}}{(2n+1)\cdot n!}$. Avec $k = 1$ et $\rho = 1$, on calcul cette fonction comme suit :

```
integrale.approxim = function(k){
  # valeur de l'expression pour n=0
  somme = 1
  # on suppose que l'infini vaut 100
  infini = 100
  # on applique la boucle for
  for (n in 1:infini){
    repartition = k**(2*n+1) / ((2*n +1)* factorial(n))
    somme       = somme + repartition
  }
  return(somme)
}
```

Application

On applique à notre fonction avec k=1 et on compare

```
resultat.2 =  integrale.approxim(1)
print(c(resultat.1$value , resultat.2))
```

```
[1] 1.462652 1.462652
```

Le résultat de gauche obtenu avec la fonction intégrée dans le logiciel et celui de droite obtenu avec notre algorithme prouvent que l'approximation de la fonction de répartition via le développement limité de Maclaurin est correcte.

Fin