Cociente absoluto

Por Álvaro Gutiérrez García

Fecha de escritura: 16/2/2022
Fecha de la última revisión: 16/2/2022

Nota introductoria: desconozco si el concepto de este operador “Cociente absoluto” que voy a describir a continuación ya es existente con otro nombre y es por ello que pido disculpas por adelantado de así serlo. Este artículo será publicado en la fecha del 5/2/2022 y surgió como solución a una necesidad matemática para desarrollar un algoritmo simple.

Introducción:

El cociente absoluto es un operador diseñado para trabajar de manera análoga al valor absoluto de una resta pero aplicada a divisiones. Esto se puede entender mejor utilizando un ejemplo.

Para una resta con un posterior valor absoluto |A-B| = C

- Si aplicamos |10 - 5| = |5| = 5
- Si aplicamos |5 - 10| = |-5| = 5

Como se puede ver, el valor absoluto nos permite obtener la diferencia (resta) entre dos valores y nos da un valor único de la distancia a diferencia de la resta tradicional. Cuando trasladamos este concepto a las divisiones para poder realizar proporciones de un valor sobre otro nos encontraremos que la división tradicional nos ofrecerá dos valores según se seleccione un valor como A y otro como B. Por ejemplo:

Para una división $\frac{A}{B} = C$

- Si aplicamos $\frac{10}{5} = 2$
- Si aplicamos $\frac{5}{10} = 0.5$

Para un mismo ratio de cociente se obtienen dos representaciones completamente distintas.

Operación básica:

Para realizar esta operación tendremos que seguir los siguientes pasos:

Para $\frac{A}{B}$ o para $\frac{B}{A}$

1) Realizar el valor absoluto a la fracción:

$$|\frac{A}{B}|$$

2) Meter la fracción en un logaritmo cuya base será completamente irrelevante , es por ello que se recomienda el uso de una base cómoda para la operación.

$$log_x(|\frac{A}{B}|)$$

3) Aplicaremos el valor absoluto al resultado de la operación:

$$|log_x(|\frac{A}{B}|)|$$

4) Finalmente, desharemos el logaritmo elevando la base al resultado obtenido:

$$x^{|log_x(|\frac{A}{B}|)|}$$

Para verlo más claro, se puede entender mejor con un ejemplo en el que tendremos los valores 10 y 5 como A y B , y utilizaremos un logaritmo base 10 (x=10):

- $10^{|log_{10}(\frac{10}{5})|} = 10^{|0.301|} = 2$
- $10^{|log_{10}(\frac{5}{10})|} = 10^{|0.301|} = 2$

Ahora, se repetirá el ejemplo con los mismos datos pero cambiando la base del logaritmo a 3 (x=3)

- $3^{|log_3(\frac{10}{5})|} = 3^{|0.630|} = 2$
- $3^{|log_3(\frac{5}{10})|} = 3^{|-0.63|} = 2$

De esta manera queda demostrada la utilidad del cociente absoluto para poder conservar una proporción entre valores sin tener que considerar cuál de los valores es mayor respecto al otro, a la vez que queda demostrado que el uso de la base logarítmica es irrelevante a la hora de calcularlo ofreciéndonos una comodidad de cálculo extra.

Como símbolo seleccionado para esta operación, he escogido:

$$\left|\frac{A}{B}\right| = x^{|log_x(|\frac{A}{B}|)|}$$

Aplicaciones , problemas y revisión:

Esta operación puede ser de gran utilidad en sectores donde se busque una comparación relativa entre dos objetos.Llevándolo al campo del tratamiento de funciones , esta operación nos podría comparar la diferencia relativa que poseen dos señales que tengan un parecido y de esta forma eliminar la redundancia para cierto umbral de manera rápida, sin embargo , esta operación ofrece una clara limitación pues no se puede aplicar a valores entre 0 y 1 , los cuales son bastante frecuentes en el mundo de las señales.

Para resolver esto bastaría con previamente realizar el valor absoluto a nuestras funciones A y B y sumarles el valor más pequeño próximo a 1, es decir $\lim_{x \to 1^+} x$ o "1.00…1"

$$x^{|log_x(\frac{|A| + \lim_{x \to 1^+}}{|B| + \lim_{x \to 1^+}})|}$$

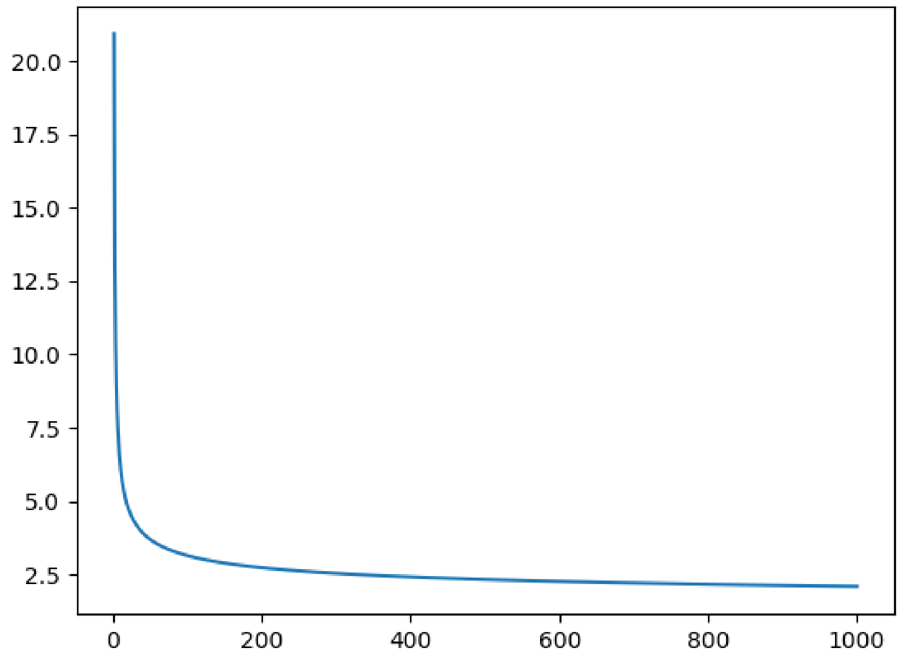
Sin embargo, este formato favorece cuando se trabaja en valores grandes como se puede observar en estos ejemplos con un cociente absoluto ideal de 2 para una base 10 y una aproximación de $\lim_{x \to 1^+} \simeq 0.001$

- A = 10 , B = 5 $\rightarrow 10^{|log_x(\frac{|10| + 0.001}{|5| + 0.001})|} = 1.9998$
- A = 10000 , B = 5000 $\rightarrow 10^{|log_x(\frac{|10000| + 0.001}{|5000| + 0.001})|} = 1.9999998$
- A = 0.1 , B = 0.05 $\rightarrow 10^{|log_x(\frac{|0.1| + 0.001}{|0.05| + 0.001})|} = 1.980$

Una forma de paliar esta imprecisión es trabajando con la operación sin realizar el último paso, es decir de manera no lineal:

$$|log_x(|\frac{A}{B}|)|$$

Si realizamos una simulación en la que repitamos la operación para unos mismos valores A y B pero variando el valor de la base utilizada para el logaritmo nos encontraremos que sus resultados se distribuyen de la siguiente manera:



Resultado obtenido con el siguiente código de python:

```
x = []
y = []
A = 10000000
B = 5
N = 1000
axis_x = np.linspace(2, N, (N-2))

for i in range(2,N):
    res = (abs(ma.log( A/B , i )))
    y.append( res)

plt.plot(axis_x,y)
plt.show()
```

Es por esta distribución que encontramos un extra de precisión en logaritmos de bases altas. Para las bases 2, 10 y 100 con A=0.1 y B = 0.05 con $\lim_{x \to 1^+} \simeq 0.001$

- $|log_2(\frac{0.1}{0.05})| - |log_2(\frac{0.1 + 0.001}{0.05+0.001})| =$ 0.01421
- $|log_{10}(\frac{0.1}{0.05})| - |log_{10}(\frac{0.1 + 0.001}{0.05+0.001})| =$ 0.00427
- $|log_{100}(\frac{0.1}{0.05})| - |log_{100}(\frac{0.1 + 0.001}{0.05+0.001})| =$ 0.002139

Como se puede comprobar, al trabajar en valores no lineales podemos encontrar ciertas diferencias (inexistentes en su modelo lineal) que nos permite reducir el error de aproximación. Hay que tener en cuenta que para trabajar con estos valores, no se puede deshacer la conversión ya que estos compartieran exactamente el mismo valor independientemente de la base utilizada.