

Tarea 2

Parte 1: Análisis del código

Cambios al código

- Para entender mejor el algoritmo, lo re-escribí al lenguaje de programación rust, usando el *crate* (librería) *nalgebra* para realizar la multiplicación de matrices.
- Aprovechando la naturaleza del algoritmo, también re-escribí *conjunto_difuso* para ser una función *branchless*, cambié su nombre a *vdif*.
- De igual forma usé *Range*<f64> en lugar de tuplas para representar a nivel del sistema de tipado que el algoritmo funciona en base a intervalos establecidos.

```
use nalgebra::Vector3;

#[inline]
fn vdif(x: f64, r: std::ops::Range<f64>) → f64 {
    1. * f64::from(x > r.end) + (x - r.start) / (r.end - r.start) * f64::from(r.contains(&x))
}

fn control_difuso(t: f64, h: f64) → f64 {
    let k_temperaturas: Vector3<_> = [0.0..15., 15.0..25., 25.0..30.0].map(|r| vdif(t, r)).into();
    let reglas_difusas: Vector3<_> = [0.0..40., 40.0..60., 60.0..100.].map(|r| vdif(h, r)).into();
    reglas_difusas.dot(&k_temperaturas)
}

fn main() {
    println!("{}", control_difuso(19., 45.));
}
```

Funcionamiento

El algoritmo utiliza dos métricas distintas, temperatura y humedad, para calcular el nivel de temperatura y humedad de forma escalar, de esto se encarga la multiplicación de matrices, de certificar que se cumple humedad y temperatura a la vez, aunque sea de forma parcial. La forma en que lo entendí, es como si se tratara de fusionar dos escalas para lograr una única.

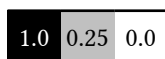
La función *control_difuso* tiene dos vectores de 3 intervalos, para una *t* y *h* dada, la puntuación será más alta mientras tanto la *t* como la *h* cubran más sectores de entre los 3 intervalos.

Por ejemplo, digamos que tenemos *let t = 22.0* y *let h = 45.0*, eso devuelve, en cada métrica individual de la función *control_difuso*:

k_temperaturas:



reglas_difusas (humedad):



En este ejemplo, es claro que la temperatura dada de *22.0* cubre totalmente el primer rango (*0..15*) de la escala de 3 posibles clasificaciones para temperatura y cubre parcialmente el segundo rango (*15..25*), más no es suficientemente grande como para cubrir el tercer rango (*25..30*), ya que *22.0 < 25.0*.

De forma similar, la humedad dada en este ejemplo (*45.0*) cubre totalmente la primera clasificación de los intervalos dados para la humedad (*0..40*), y parcialmente la segunda (*40..60*).

Es por ello que el algoritmo devuelve que se sabe con certeza que el primer cuadrante se satisface tanto para humedad como temperatura (valor de 1.), y que el segundo cuadrante es parcial, por lo que obtiene $0.7 * 0.25$ para tener un valor único representativo, por lo que el algoritmo arroja el escalar resultante $1 + (0.7 + 0.25) = 1.175$.

Lo que podría verse, de forma gráfica como:

resultado = 1.175:

1.0	0.18	0.0
-----	------	-----

Como son 3 clasificaciones posibles para humedad y temperatura el valor máximo que puede devolver el algoritmo es 3.0, que significa que la tupla (t, h) es suficientemente grande como ser parte de las 3 clasificaciones dadas y cubrirlas por completo.

Por la multiplicación de matrices que se realiza, si solo se cubre temperatura, por poner un ejemplo, más no humedad, entonces el resultado final refleja que no se están cumpliendo con las disitintas categorías, por ejemplo: `let t = 100.0` y `let h = 10.0`:

k_temperaturas:

1.0	1.0	1.0
-----	-----	-----

reglas_difusas (humedad) :

0.25	0.0	0.0
------	-----	-----

resultado = 0.25:

0.25	0.0	0.0
------	-----	-----

Es decir, que la tupla (t, h) solo cumple, parcialmente con las condiciones necesarias del primer intervalo de clasificación, cuando digo tupla trato de hacer énfasis en que me refiero a las condiciones de los dos valores en simultaneo.

Parte 2: Leyes de Conjuntos

$$1. A' \cap B' \cap C \cup A \cap B' \cap C' \cup A \cap B' \cap C \cup A \cap B \cap C \cup A \cap B \cap C' = A \cup B' \cap C$$

$$\begin{aligned} &= (A' \cap B' \cap C) \cup (A \cap B' \cap C') \cup (A \cap B' \cap C) \cup (A \cap B \cap C) \cup (A \cap B \cap C') \\ &= (A' \cap B' \cap C) \cup (A \cap B' \cap C') \cup (A \cap B' \cap C) \cup (A \cap B) \cap (C \cup C') \\ &= (A' \cap B' \cap C) \cup (A \cap B') \cap (C' \cup C) \cup (A \cap B) \\ &= (A' \cap B' \cap C) \cup A \cap (B' \cup B) \\ &= (A' \cap B' \cap C) \cup A \\ &= (A \cup A' \cap A \cup B' \cap A \cup C) \\ &= A \cup B' \cap A \cup C \\ &= A \cup B' \cap C \end{aligned}$$

$$2. A' \cap B' \cap C \cup A' \cap B \cap C \cup A' \cap B \cap C' \cup A \cap B' \cap C \cup A \cap B \cap C \cup A \cap B \cap C' = B \cup C$$

$$\begin{aligned} &= (A' \cap B' \cap C) \cup (A' \cap B \cap C) \cup (A' \cap B \cap C') \cup (A \cap B' \cap C) \cup (A \cap B \cap C) \cup (A \cap B \cap C') \\ &= (A' \cap B' \cap C) \cup (A' \cap B) \cap (C \cup C') \cup (A \cap B' \cap C) \cup (A \cap B) \cap (C \cup C') \\ &= (A' \cap B' \cap C) \cup (A' \cap B) \cup (A \cap B' \cap C) \cup (A \cap B) \\ &= (A' \cap B' \cap C) \cup (A \cap B' \cap C) \cup B \cap (A \cup A') \\ &= (A' \cap B' \cap C) \cup (A \cap B' \cap C) \cup B \\ &= (B' \cap C) \cap (A' \cup A) \cup B \\ &= B' \cup B \cap C \cup B \\ &= C \cup B \end{aligned}$$