01_Assignment_01

March 27, 2022

1 Fuzzy Logic and Soft Computing (LTAT.02.005)

2 ASSIGNMENT 1

Student: ChengHan Chung

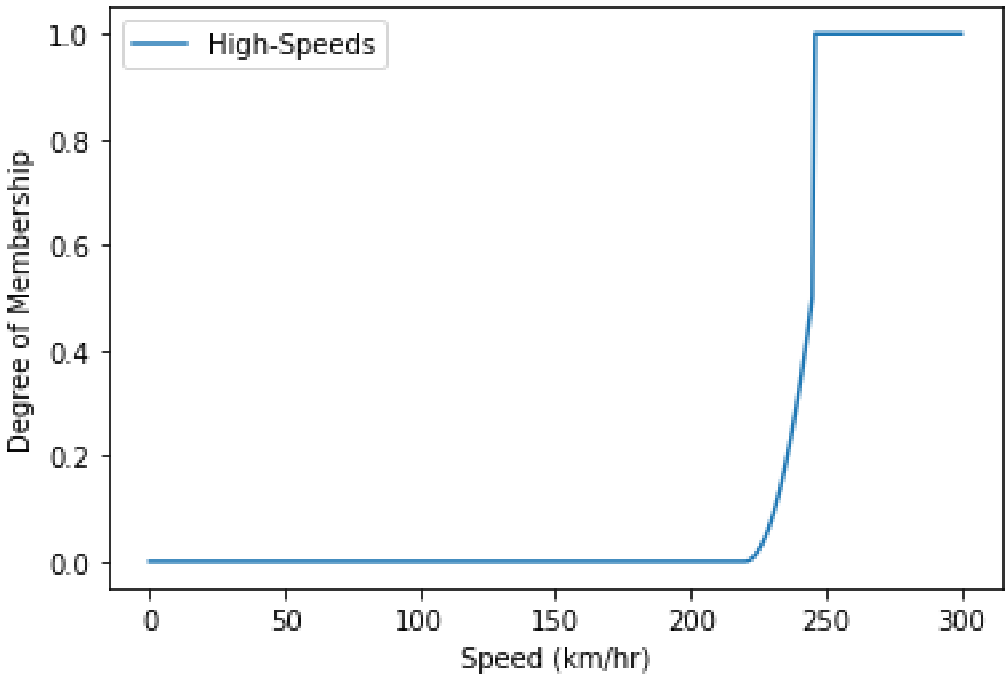
---

1. Consider the fuzzy set of high speeds for racing cars.

   a. Model it properly, by plotting the membership function and commenting on your choice (2 points).

   (A) : We have a Fuzzy set $A = [0, 300]$, the min speed is `0`, and max speed is `300`. If spped is lower than 220, then the $\mu = 0$. If speed is greater than 220 but lower than 245, the $\mu = \frac{1}{2}(\frac{x-220}{25})^2$. If the speed is greater than 245, then the $\mu = 1$. The `membership function` $\mu_A(X)$ shown as below:

$$\mu_A(x) = \begin{cases} 0, & X \leq 220 \\ \frac{1}{2}(\frac{x-220}{25})^2, & 220 < x \leq 245 \\ 1, & x > 245 \end{cases}$$



b. Determine the support, core and height of the fuzzy set above (2 points).

(A) :

Support is the set of $\mu > 0$, therefore, the $support = \{(220, 300]\}$.

Core is the set of $\mu = 1$, hence, the $core = \{(245, 300]\}$.

Heigh is the max value of μ, so the heigh $h(A) = max_{x \in X} A(x) = 1$.

c. Show and comment on the intersection of the fuzzy set above and the fuzzy set of non-high speeds (1 point).

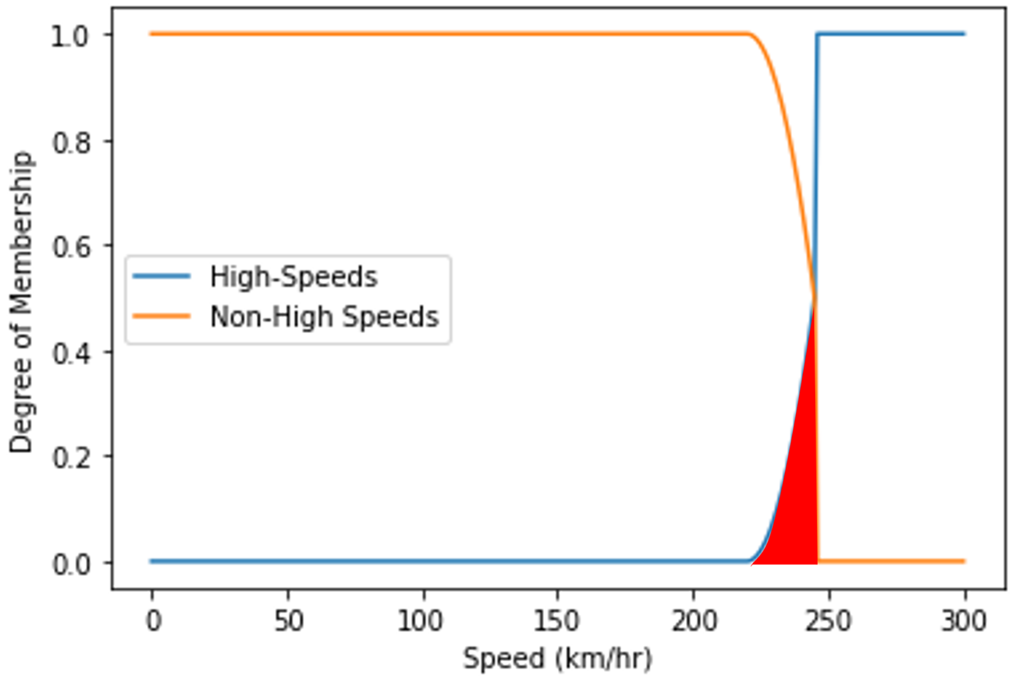
(A) : Assume we have the fuzzy set of high speeds A and non-high speeds B (complement of high speeds), the intersection of both fuzzy set is:

$D = A \cap B$, $\mu_D(x) = min\{\mu_A(x), \mu_B(x)\}, x \in X$

,Hence:

$$min\{\chi_A(x), \chi_B(x)\} = \begin{cases} 1, & if\ x \in A\ and\ x \in B \\ 0, & if\ x \notin A\ or\ x \notin B \end{cases} = \begin{cases} 1, & if\ x \in A \cap B \\ 0, & if\ x \notin A \cap B \end{cases} = \chi_{A \cap B}(x), \quad x \in X$$

The intersection of high speeds A and non-high speeds B shown as following below(red area):



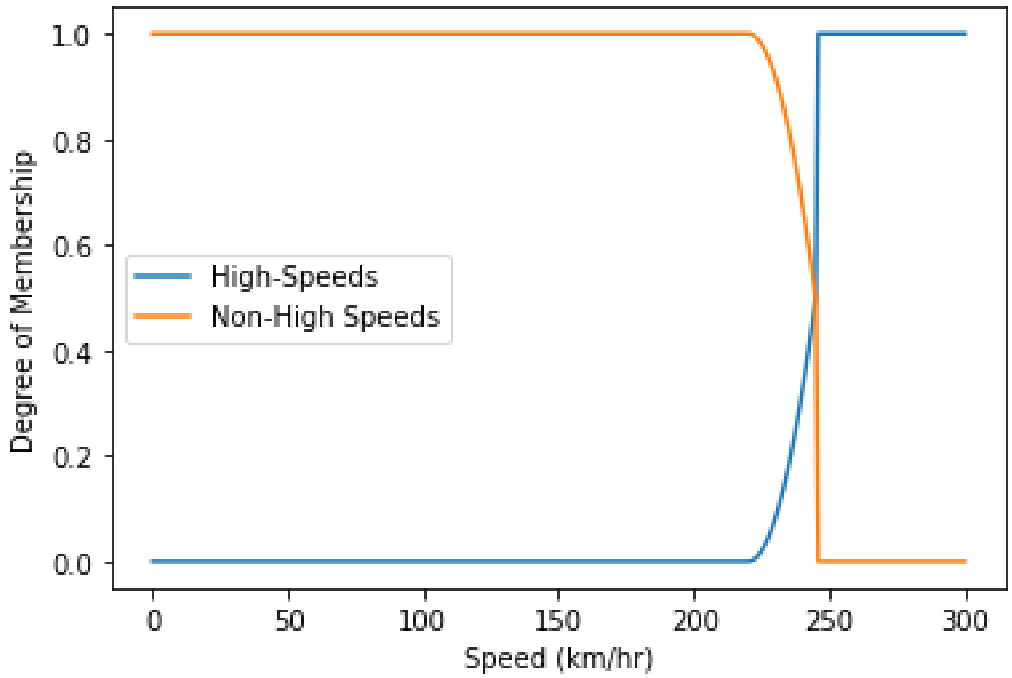
[11]:
```
import matplotlib.pyplot as plt
```

[12]:
```
# Create a speed list in a range of 0-300
speed_set = [*range(0, 301, 1)]
fuggy_set = []
for speed in speed_set:
    if speed <= 220:
        fuggy_set.append(0)
    elif 220 < speed <= 245:
        fuggy_set.append(0.5*(((speed-220)/25)**2))
    else:
        fuggy_set.append(1)

intersection_set = [(1-fs) for fs in fuggy_set]

plt.plot(fuggy_set,label="High-Speeds")
plt.plot(intersection_set,label='Non-High Speeds')

plt.ylabel('Degree of Membership')
plt.xlabel('Speed (km/hr)')
plt.legend()
plt.show()
```



2. Write a pseudocode for the addition of two finite fuzzy numbers by the Extension principle (6 points). Note: in case you prefer it, the pseudocode can be replaced by a Scilab code.

(A) :

```
A1[(mu_x1i,x1_i)] ← Fuzzy Numbers
A2[(mu_x2i,x2_i)] ← Fuzzy Numbers

TWO_FUZZY_NUMBERS_ADDITION(y,A1,A2):
    B[] ← Fuzzy Set
    for i range(len(A1)) do
        if A1[i].mu_x1i == 0 then
            skip
        end if
        for j range(len(A2)) do
            if A2[j].mu_x2i == 0 then
                skip
            end if
            if (A1[i].x1_i + A2[j].x2_i) == y then
                B.append(MIN(A1[i].mu_x1i,A2[j].mu_x2i))
            end if
        end for
    end for

    return MAX(B)
```

3. Show an example of a set-relation composition with a finite fuzzy set representing *medium* and a fuzzy relation *much smaller than.* Elaborate on the outcome (see also section 3.2.1 textbook). You can use the code shared in Practice 3 to get the numerical result (3 points).

   (A) : Assume we have a fuzzy set A representing *medium* and a fuzzy relation B *much smaller than.* and the prrduct of result is shown as following below:

$$A = \begin{bmatrix} .1 & .2 & .3 \\ .4 & .5 & .6 \end{bmatrix} B = \begin{bmatrix} .7 & .8 & .9 \\ .1 & .2 & .3 \\ .4 & .5 & .6 \end{bmatrix}$$

```
function C = max_min(A, B)
//this function returns the max-min composition of given matrices.

[m,n] = size(A);
[p,q] = size(B);
if (n ~= p) error('The matrices have incompatible sizes.'); end

C = zeros(m,q);
for i = 1:m
    for j = 1:q
                tm= min(A(i, :), B(:, j)');
            C(i,j) = max(tm);
        end
    end
endfunction
```

```
Startup execution:
  loading initial environment

--> exec('C:\Users\chenghan\Documents\max_min.sce',-1)

--> A=[0.1,0.2,0.3;0.4,0.5,0.6]
 A  =

   0.1   0.2   0.3
   0.4   0.5   0.6

--> B=[0.7,0.8,0.9;0.1,0.2,0.3;0.4,0.5,0.6]
 B  =

   0.7   0.8   0.9
   0.1   0.2   0.3
   0.4   0.5   0.6

--> C=max_min(A,B)
 C  =

   0.3   0.3   0.3
   0.4   0.5   0.6

-->
```