

Visa : Fusion de données
TP 1 : Approche de la logique floue

Introduction	1
Fonctions d'appartenance	1
Génération de courbes d'appartenances	1
Exemple : calcul de degré d'appartenance	2
Calcul d'union d'ensemble	2
II. Opérateurs de la logique floue	4
L'opérateur min	4
L'opérateur max	5
III. Implication floue	7

Introduction

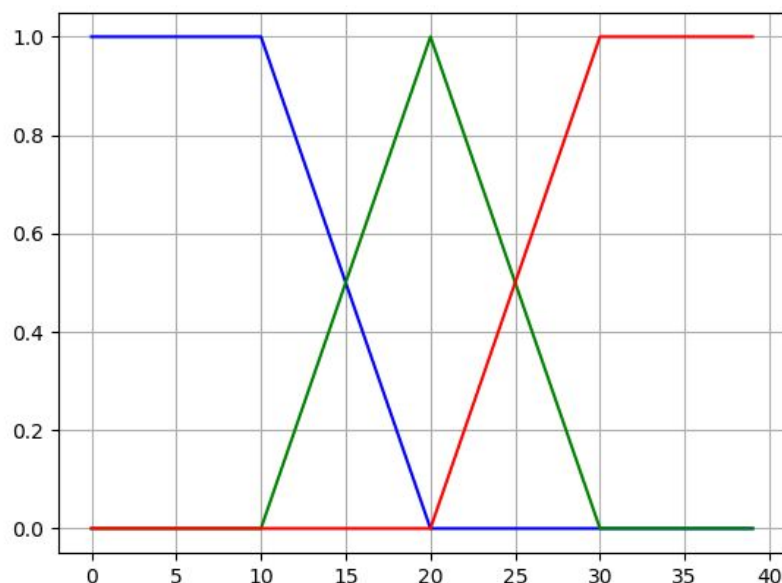
L'objectif de ce TP est de se familiariser avec les concepts de base de la logique flou. Pour ce faire nous étudierons les principes de fonctions d'appartenance, les opérateurs de la logique flou ainsi qu'un exemple d'utilisation de l'implication de Mamdani. Les manipulations seront effectués en python, seul l'essentiel du code sera expliqué. L'ensemble du code se trouve joint à ce rapport.

La logique floue est une logique polyvalente où contrairement à la logique booléenne, les valeurs de vérité sont des variables comprises entre 0 et 1 et non pas vrai ou faux. Ainsi, les valeurs de vérité sont représenté par un degré d'appartenance plus ou moins élevé. L'univers du discours désigne l'ensemble des entités parcouru par les quantificateurs.

I. Fonctions d'appartenance

1. Génération de courbes d'appartenances

Considérons une partition floue d'un univers du discours associé à une variable numérique *Température*. Nous décidons de tracer trois courbe représentant la classification flou des valeurs de température. Ces trois courbes représentent les températures "Basses" en bleu, "Moyennes" en verte et "Élevées" en rouge.



*Figure 1 : Fonctions d'appartenance des températures basses, moyennes et élevées.
En abscisse les température et en ordonnées les degrés de vérités*

Pour générer ces courbes nous utilisons une fonction renvoyant une valeur pour une température passé en paramètre. Par exemple la fonction générant l'appartenance aux basses températures, la méthode est similaire pour les autres courbes.

```
# Retourne pour une valeur temp l'appartenance à la classe basse
température
def basse_temp(temp):
    # SI La température est inférieur à 10
    if(temp <= 10):
        # ALORS la température est basse
        return 1
    # SI la température est supérieur à 20
    if(temp >= 20):
        # ALORS la température n'est pas basse
        return 0
    # SI la température est compris entre 10 et 20
    # ALORS son appartenance à température basse est comprise entre 0 et
    # 1
    return float(20 - temp)/float(20 - 10)
```

Ce calcul d'appartenance nous permet de représenter le principe des règles de la logique flou : *“Si la température est inférieur à 10 ALORS elle est basse”* ou encore *“Si la température est comprise entre 10 et 20 ALORS elle est moyennement basse”*

On itère ensuite cette méthode sur des températures comprises entre 0 et 40 avec plus ou moins de précisions (ici 400 valeurs de 0 à 40 avec un pas de 0,1).

```
# Pour 400 valeur de 0 à 40
for i in range(0,400):
    # De 0 à 40°C
    temp.append(i/10)
    # basse contient les degrés d'appartenance de la classe basse
    # de chaque température
    basse.append(basse_temp(i/10))
```

2. Exemple : calcul de degré d'appartenance

Grâce à ses courbes nous pouvons classier les différentes températures. Par exemple, pour une température de 16°C nous pouvons graphiquement et par le calcul définir les différents degrés d'appartenances :

```
kasperek@b02p14:~/Documents/M2/VISA/fuzzy_tp1$ python Ensembles.py
Pour 16C :
('Appartenance basse :', 0.4)
('Appartenance moyenne :', 0.6)
('Appartenance haute :', 0)
```

- Appartenance température basse : 0,4

- Appartenance température moyenne : 0,6
- Appartenance température haute : 0,6

3. Calcul d'union d'ensemble

Comme pour la logique booléenne il est possible de réaliser des unions d'ensembles afin de définir de nouvelle possibilité de degré d'appartenance. Nous décidons de calculer et de représenter la courbe d'appartenance de l'ensemble flou "Température basse ou moyenne".

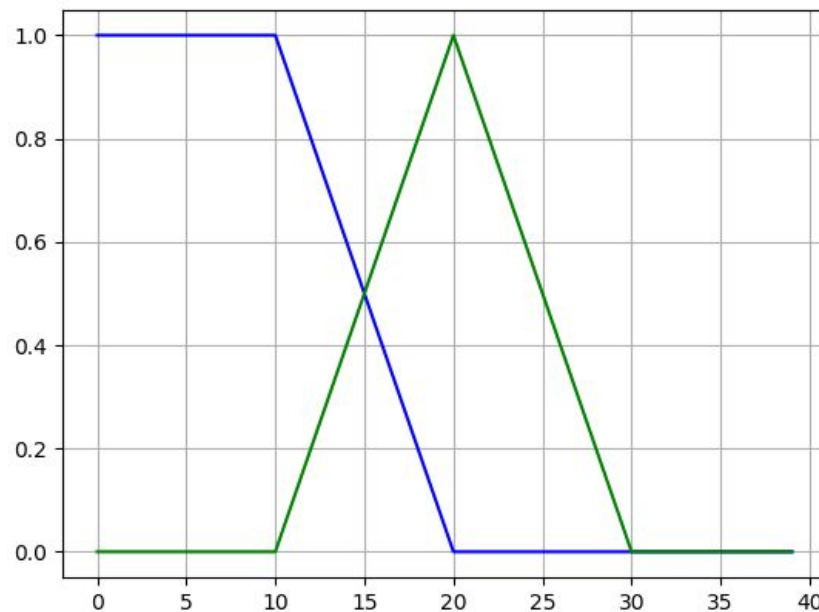


Figure 2 : Fonctions d'appartenance des températures basses et moyennes. En abscisse les température et en ordonnées les degrés de vérités

Nous obtenons alors la courbe :



Figure 3 : Courbe représentant l'union des températures basses et moyennes. En abscisse les température et en ordonnées les degrés de vérités

Nous pouvons remarquer que cette courbe correspond à la valeur maximum entre les deux courbes précédentes.

II. Opérateurs de la logique floue

Dans la partie précédente nous avons exploré les bases de la logique floue à travers les fonctions d'appartenance. Nous avons fini par étudier le cas d'union d'ensemble et noter qu'il correspond à un calcul de maximum entre les fonctions d'appartenance. Dans cette partie nous implémenterons et présenterons les résultats des opérateurs minimum et maximum.

1. L'opérateur min

Pour implémenter un opérateur minimum sur les fonctions d'appartenances nous écrivons la fonction *fuzzy_min* ci-dessous. Il suffit de faire le minimum entre chaque valeur de vérité au même indice :

```
def fuzzy_min(ens1, ens2):  
    min_t = []  
    for i in range(0, len(ens1)):  
        min_t.append(min(ens1[i], ens2[i]))  
    return min_t
```

Voici quelques exemples d'utilisation de la fonction *fuzzy_min* sur les différents ensembles. min basse haute

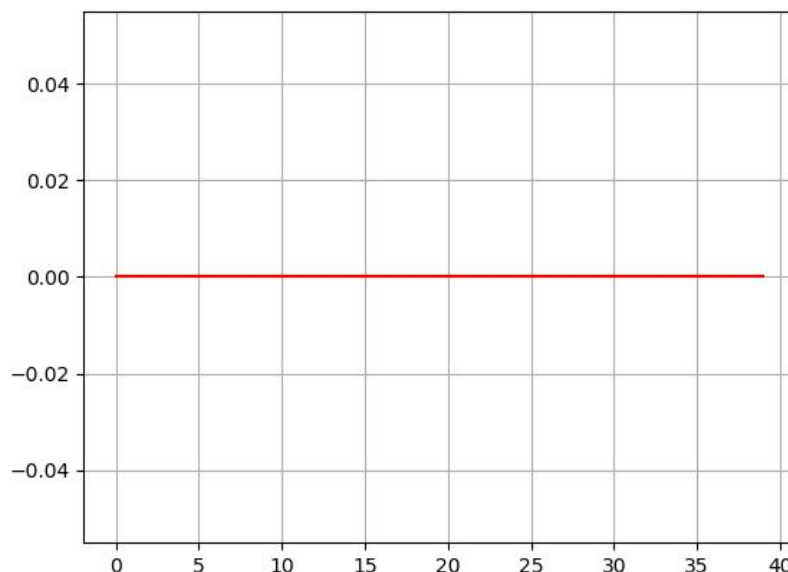
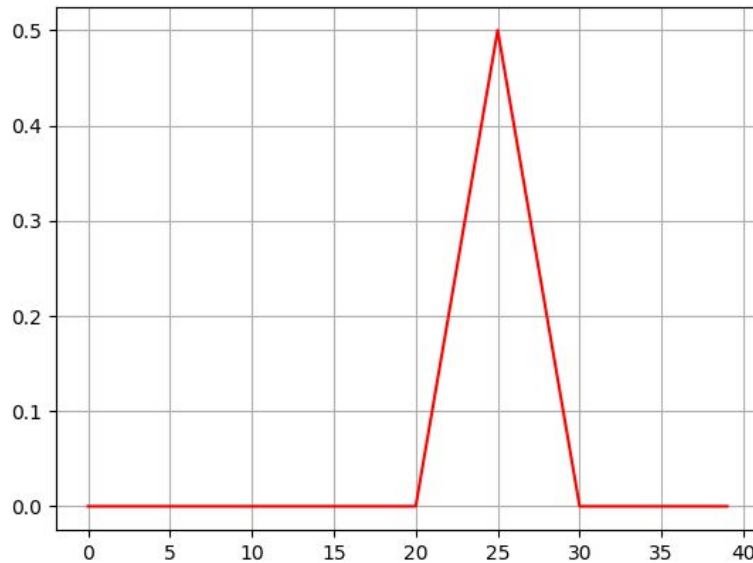


Figure 4 : Courbe représentant le minimum des températures basses et élevées. En abscisse les température et en ordonnées les degrés de vérités

Cette courbe est une constante de 0 puisque les deux ensembles n'ont pas de plage de valeur en communs différentes de zéros. Ainsi nous remarquons qu'une température ne peut pas être basse et élevée à la fois.



*Figure 5 : Courbe représentant le minimum des températures moyennes et élevées.
En abscisse les température et en ordonnées les degrés de vérités*

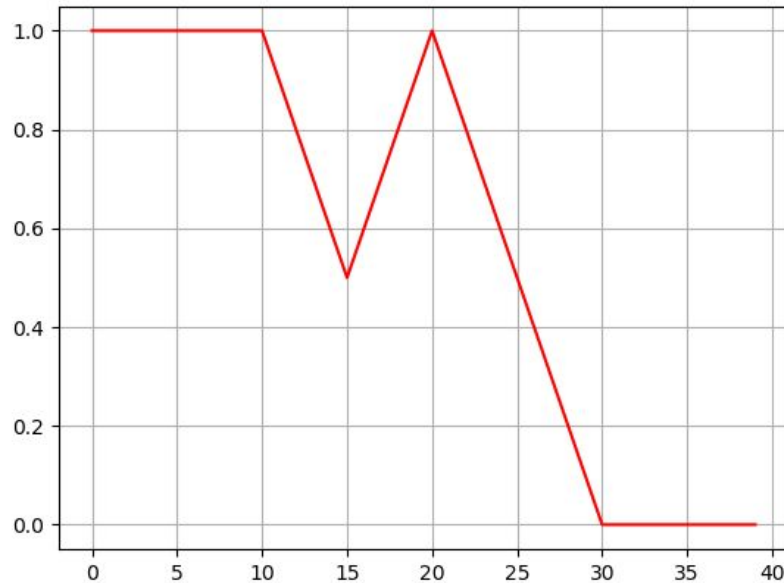
Dans cette seconde courbe, nous remarquons un pic entre les températures 20 et 30. Ces valeurs correspondent aux températures moyennement élevée avec un maximum à 0,5.

2. L'opérateur max

De la même manière la fonction `fuzzy_max` réalise l'opérateur maximum entre deux ensembles :

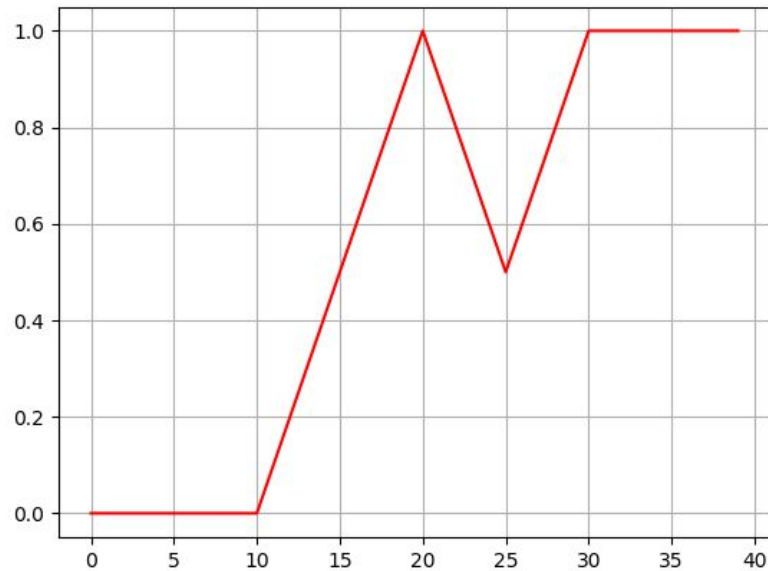
```
def fuzzy_max(ens1, ens2):  
    max_t = []  
    for i in range(0, len(ens1)):  
        max_t.append(max(ens1[i], ens2[i]))  
    return max_t
```

Ci-dessous des exemples de courbes générés grâce au maximum entre différents ensembles :



*Figure 6 : Courbe représentant le maximum des températures basses et moyennes.
En abscisse les température et en ordonnées les degrés de vérités*

La courbe ci-dessus représente le maximum entre l'ensemble basse et moyenne température. Nous pouvons constater que cela correspond bien à l'ensemble "basse ou moyenne température" calculé dans la partie précédente.



*Figure 7 : Courbe représentant le minimum des températures moyennes et élevées.
En abscisse les température et en ordonnées les degrés de vérités*

Ainsi, la courbe ci-dessus représente quant à elle le maximum entre la courbe des moyennes et celle des hautes températures. Cette fonction d'appartenance correspond au degré d'appartenance des températures à la classe "température moyenne ou élevé".

III. Implication floue

Nous souhaitons à présent appliquer ces propriétés à un exemple concret. Nous calculons le sous-ensemble flou “Chauffer fort”.



Figure 8 : Courbe représentant l'ensemble “Chauffer fort”. En abscisse les température et en ordonnées les degrés de vérités

Soit une température mesurée de 12°C. Nous cherchons à représenter l'ensemble flou de sortie correspondant à la règle flou R “Si température basse ALORS chauffer fort” grâce à l'implication de Mamdani. Pour cela nous allons réaliser une opération minimum entre la valeur de vérité de température basse pour 12°C et la courbe “chauffer fort”.

```
# Pour 400 valeur de 0 a 40
for i in range(0,400):
    # De 0 a 40C
    temp.append(i/10)
    # basse contient les degres d'appartenance de la classe basse
    # de chaque temperature
    basse.append(basse_temp(i/10))
    moy.append(moy_temp(i/10))
    haute.append(haute_temp(i/10))
    chauffe.append(fuzzy_chauffe(i/10))
    basseOrMoy.append(max(basse_temp(i/10),moy_temp(i/10)))
```

```

for i in range(0,400):
    # Courbe constante au degré de vérité de température basse à 12°
    constante.append(basse_temp(12))
    # Trace les courbes constante, puissance de chauffe et le min des deux
    plt.plot(temp, constante, 'b')
    plt.plot(temp, chauffe, 'g')
    plt.plot(temp, fuzzy_min(constante,chauffe), 'r')

```

Nous pouvons observer en bleu la valeur de vérité de température basse pour 12°C, soit 0.8, en vert le sous ensemble “Chauffer fort”. En rouge la courbe représentant le minimum entre les deux ensembles.

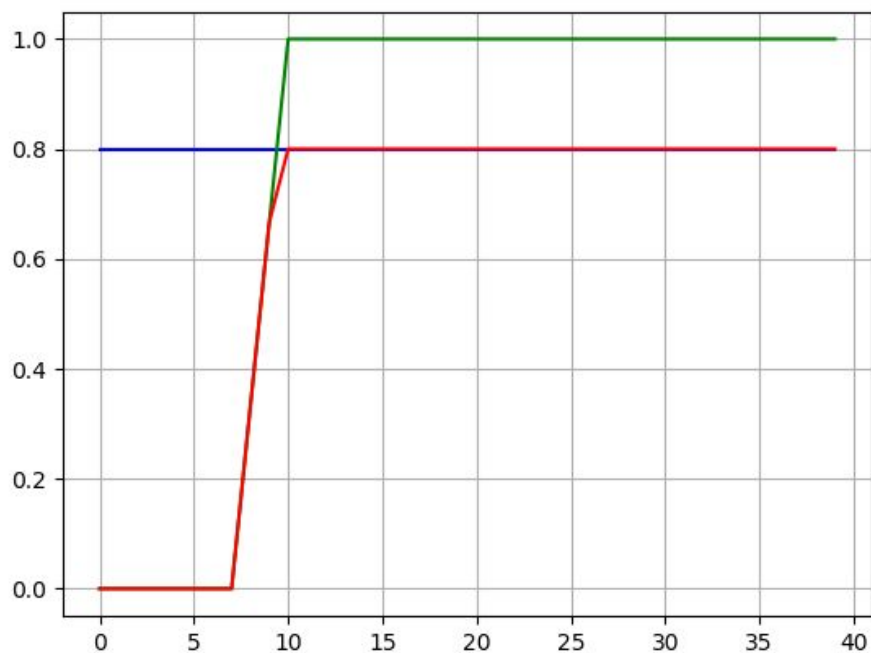


Figure 8 : En abscisse les température et en ordonnées les degrés de vérités

Conclusion

Dans ce TP nous avons exploré les principes de fonctions d'appartenances, d'opérateurs sur la logique floue ainsi qu'un exemple d'utilisation de l'implication de Mandami. Nous avons pu nous familiariser avec les concepts de base de la logique floue afin de comprendre comment obtenir des résultats en fonction de règles données.