

VISA – TP1 : Fuzzy Logic

Arnaud Cojez

mardi 18 octobre 2016

Table des matières

1	Introduction	3
1.1	Motivation	3
1.2	Explication	3
2	Fonctions d'appartenance	4
2.1	Présentation du problème	4
2.1.1	Explication	4
2.1.2	Implémentation Python	4
2.2	Degré d'appartenance	5
2.2.1	Explication	5
2.2.2	Implémentation Python	5
2.2.3	Trace	6
2.3	Température basse ou moyenne	6
2.3.1	Explication	6
2.3.2	Implémentation Python	6
3	Opérateurs de la logique floue	7
3.1	Explication	7
3.2	Opérateur min	7
3.2.1	Explication	7
3.2.2	Implémentation Python	7
3.3	Opérateur max	8
3.3.1	Explication	8
3.3.2	Implémentation Python	8
4	Implication floue	9
4.1	Règle floue R	9
4.1.1	Explication	9
4.1.2	Implémentation Python	9
4.2	Implication de Mamdani	10
4.2.1	Explication	10
4.2.2	Implémentation Python	10
5	Conclusion	11

1 Introduction

1.1 Motivation

Dans le domaine de la reconnaissance de formes, la logique booléenne peut ne pas être suffisante. En effet, certains pixels sembleront appartenir à plusieurs zones d'une même image.

Afin de définir à quelle zone appartient tel pixel, on utilise la logique floue.

1.2 Explication

Le principe de la logique floue est de donner un pourcentage d'appartenance pour chaque zone, et non pas de dire simplement si un pixel appartient à une zone ou non.

Nous allons expliquer ce concept et la démarche à suivre le long de ce rapport.

2 Fonctions d'appartenance

2.1 Présentation du problème

2.1.1 Explication

Nous sommes en présence d'une partition floue correspondant aux sous-ensembles flous suivants : Température Basse, Moyenne, Haute, associés à une variable Température (en °C) (Figure ci-dessous).

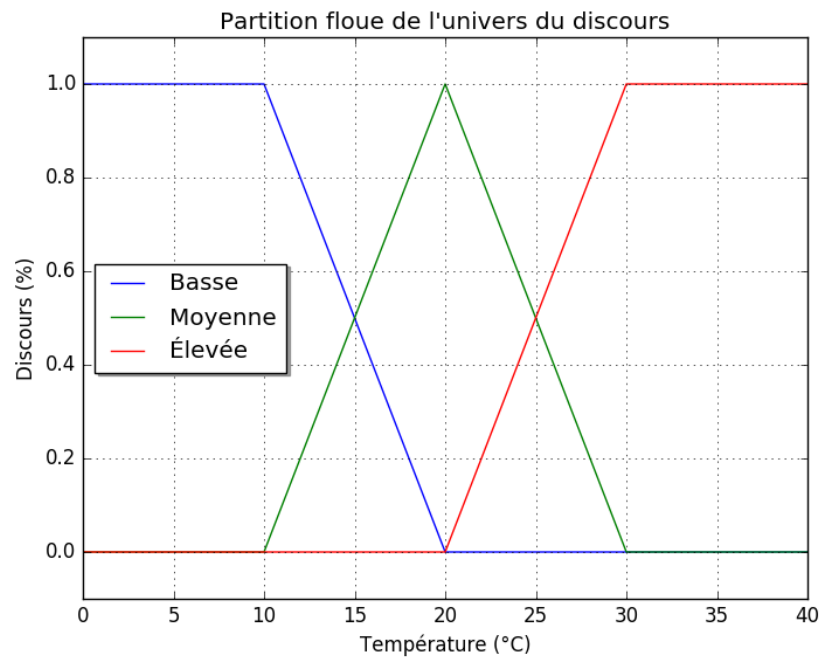


FIGURE 1 – Discours associé à la température en °C

Pour obtenir de telles courbes, il nous a fallu définir les fonctions ci-dessous.

2.1.2 Implémentation Python

```
1 def basse(t):  
2     return np.clip((-t + 20.) * 0.1, 0., 1.)  
3  
4 def moyenne(t):  
5     return 1. - np.clip(np.sign(t - 20) * (t - 20) * 0.1,  
6         0., 1.)  
7  
8 def elevee(t):  
9     return np.clip((t - 20.) * 0.1, 0., 1.)
```

2.2 Degré d'appartenance

2.2.1 Explication

Maintenant que nous avons les fonctions modélisant les différents discours, nous pouvons définir le discours associé à une température mesurée. Ici nous prenons l'exemple de 16°C.

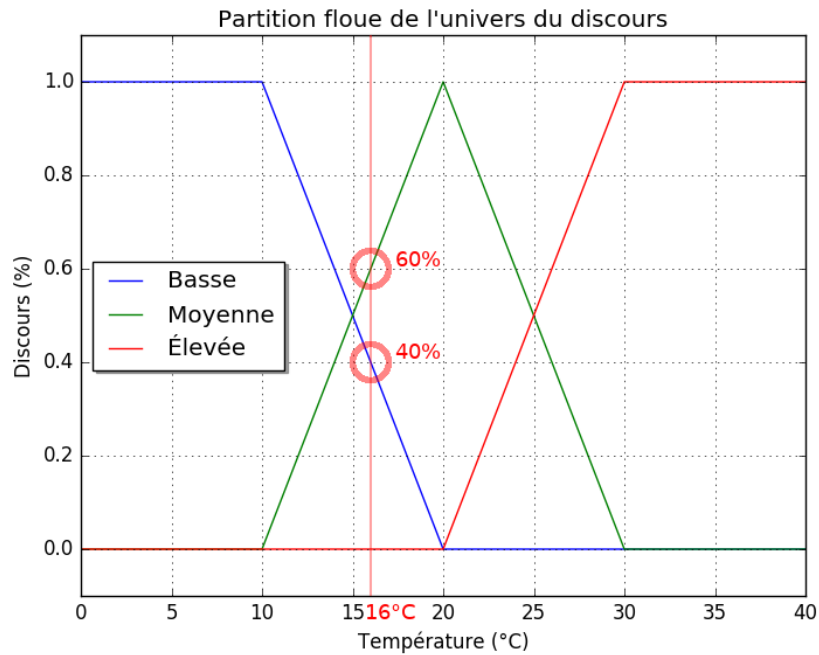


FIGURE 2 – Appartenance pour une température de 16°C

Nous voyons que la température de 16°C appartient à 40% au discours "Température Basse" et à 60% au discours "Température Moyenne", ce qui laisse apparaître une certaine transition dans le discours, transition qui n'est pas possible avec la logique booléenne.

Pour ce faire, nous avons besoin d'implémenter les fonctions et procédures suivantes :

2.2.2 Implémentation Python

```
1 def get_appartenance(t):  
2     return [basse(t), moyenne(t), elevee(t)]  
3  
4 def print_appartenance(t):  
5     l = get_appartenance(t)  
6     print("Basse = ", l[0])  
7     print("Moyenne = ", l[1])  
8     print("Elevee = ", l[2])
```

2.2.3 Trace

Voici le résultat obtenu à l'exécution :

```
1 Question 2 : Appartenance de 16°C
2 Basse = 40.0 %
3 Moyenne = 60.0 %
4 Elevee = 0.0 %
```

2.3 Température basse ou moyenne

2.3.1 Explication

Nous pouvons utiliser des opérations logiques sur les ensembles de logique floue.

Ci-dessous nous avons pour exemple l'ensemble flou "Température Basse OU Moyenne".

Dans ce cas, l'opérateur OU peut-être remplacé par une fonction renvoyant le maximum (De même, ET sera remplacé par la fonction minimum).

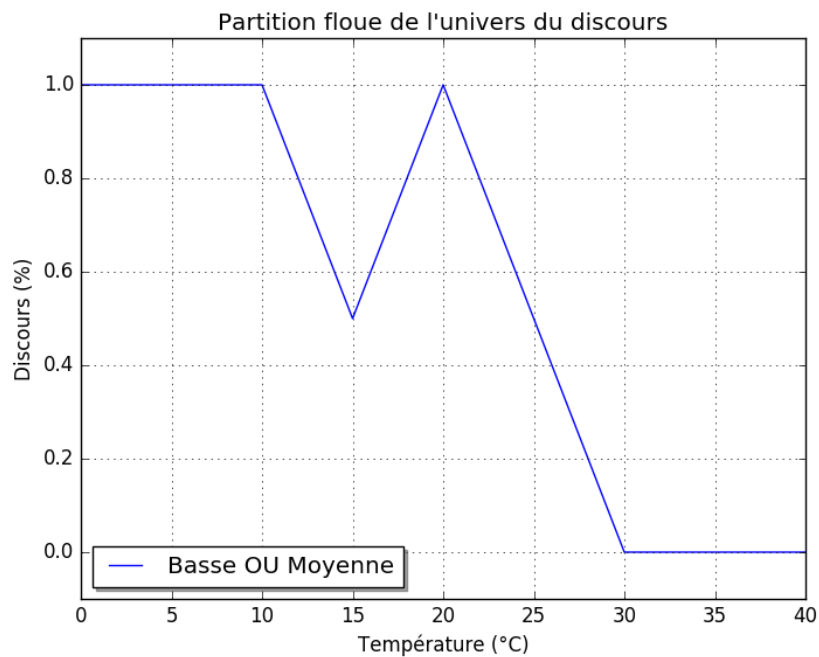


FIGURE 3 – Température basse OU moyenne

2.3.2 Implémentation Python

```
1 def basse_ou_moyenne(t):
2     return np.maximum(basse(t), moyenne(t))
```

3 Opérateurs de la logique floue

3.1 Explication

Nous pouvons également implémenter différents opérateurs afin d'obtenir le sous-ensemble flou résultant d'opérations effectuées sur d'autres sous-ensembles. Nous allons ici implémenter les opérateurs min et max.

3.2 Opérateur min

3.2.1 Explication

L'opérateur min renvoie le plus petit des $f_i(x)$, pour chaque fonction f_i correspondant aux sous-ensembles d'entrées.

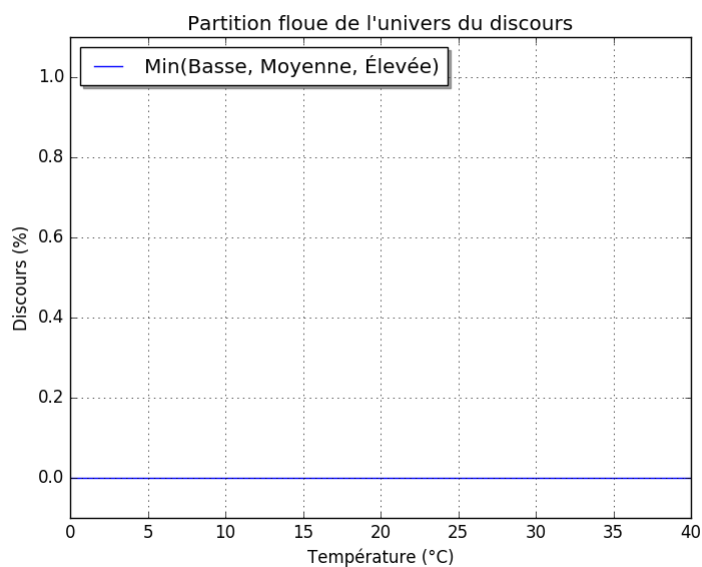


FIGURE 4 – min(Basse, Moyenne, Élevée)

3.2.2 Implémentation Python

```
1 def op_min(fs, t):
2     res = []
3     for f in fs:
4         if res == []:
5             res = f(t)
6         else :
7             newRes = np.minimum(res, f(t))
8             res = newRes
9     return res
```

3.3 Opérateur max

3.3.1 Explication

L'opérateur max renvoie le plus grand des $f_i(x)$, pour chaque fonction f_i correspondant à un des sous-ensembles d'entrées.

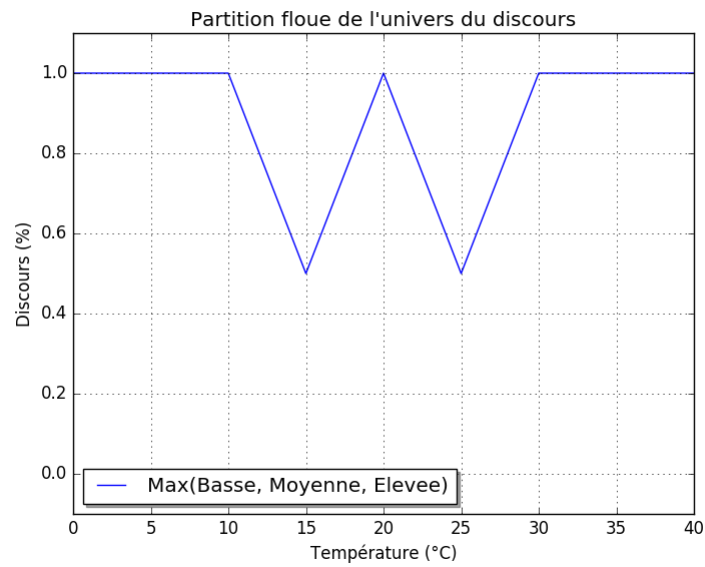


FIGURE 5 – max(Basse, Moyenne, Élevée)

3.3.2 Implémentation Python

```
1 def op_max(fs, t):
2     res = []
3     for f in fs:
4         if res == []:
5             res = f(t)
6         else :
7             newRes = np.maximum(res, f(t))
8             res = newRes
9     return res
```


4 Implication floue

4.1 Règle floue R

4.1.1 Explication

Soit la règle floue R : Si «Température Basse» ALORS «Chauffer Fort».
Ci-dessous nous voyons la représentation du sous-ensemble «Chauffer Fort».

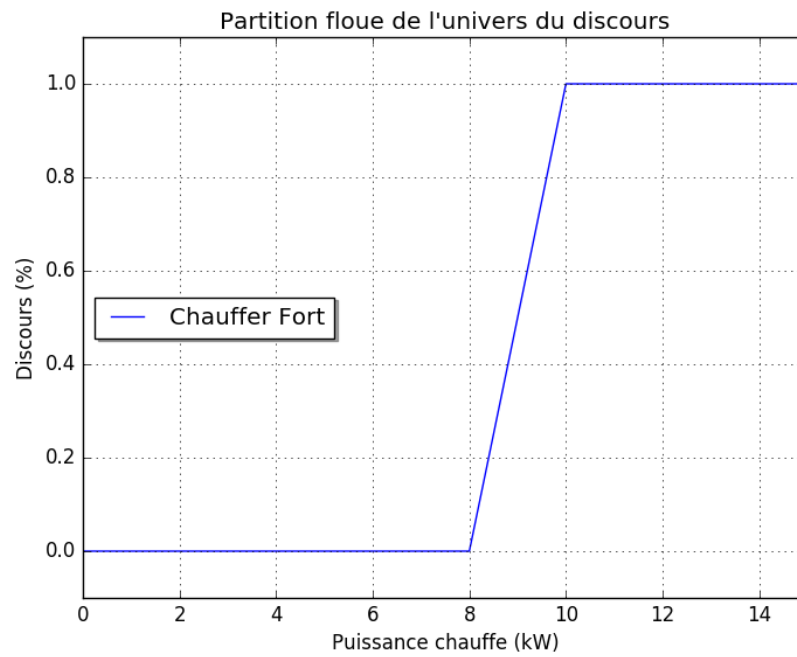


FIGURE 6 – Discours «Chauffer fort» en fonction de la Puissance de Chauffage

Pour obtenir ce sous-ensemble flou, nous avons dû implémenter la fonction suivante.

4.1.2 Implémentation Python

```
1 def chauffer_fort(t):  
2     return np.clip((t - 8.) * 0.5, 0., 1.)
```

4.2 Implication de Mamdani

4.2.1 Explication

Afin de calculer l'implication relative à la règle R citée plus haut, nous allons utiliser la méthode de Mamdani :

$$\forall y \in U_y, \mu'_{conclusion}(y) = \min_{y \in U_y} (\mu_{predicate}(x_0), \mu_{conclusion}(y)) \quad (1)$$

Cette méthode ainsi que l'application de la formule nous permettent d'obtenir le sous-ensemble suivant, représentant la règle R pour une température de 12°C :

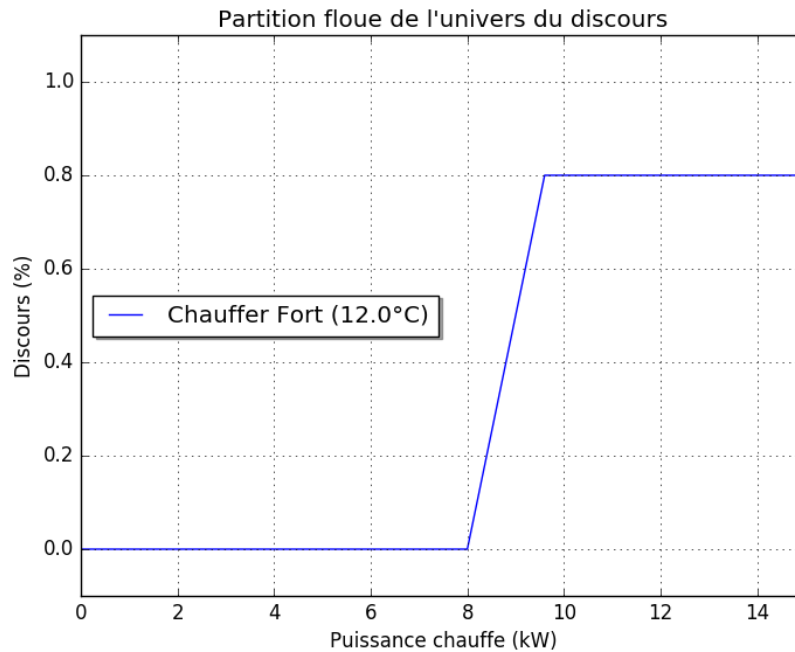


FIGURE 7 – Implication de Mamdani Température/Chauder pour 12°C

4.2.2 Implémentation Python

```
1 def mamdani(predicate, x0, conclusion, y):
2     return np.minimum(predicate(x0), conclusion(y)))
3
4 #[...]
5 # Appel de la fonction mamdani :
6     mamdani(basse, temp, chauffer_fort, t)
7 #[...]
```

5 Conclusion

Nous avons vu dans ce rapport comment appliquer les concepts de logique floue, ainsi que la méthode de Mamdani.

Nous avons constaté l'intérêt de ce type de modélisation. En effet, nous pouvons représenter des transitions entre différents états ou discours, ce qui n'était pas possible avec la logique classique, ou booléenne.

Nous avons également pu voir une des applications pratiques de la logique floue : le changement de la puissance de chauffe en fonction de la température. Ce ne sont pas les seules applications, la logique floue est également utilisée en traitement d'image afin de reconnaître des zones et des éléments, dans les stabilisateurs des caméras, ou encore dans le métropolitain, afin de gérer l'accélération de la rame, entre autres.