

Modélisation de l'incertitude — Master 2 MIAGE IA²

Travaux dirigés N° 6 : Logique floue

Andrea G. B. Tettamanzi
Université côte d'Azur
andrea.tettamanzi@univ-cotedazur.fr

Année universitaire 2020/2021

Résumé

L'objectif de ce TP est de réaliser un système d'inférence flou de type Mamdani dont la base contient six règles. Pour ce faire, on s'appuiera sur la bibliothèque Python `skfuzzy` (cf. <https://scikit-fuzzy.readthedocs.io/en/latest/>).

1 Introduction

La bibliothèque Python `skfuzzy` peut être installée avec la commande

```
pip install -U scikit-fuzzy
```

Lisez le guide utilisateur sur la page Web de `skfuzzy` pour vous familiariser avec cette bibliothèque, dont l'utilisation est assez intuitive.

2 Système d'inférence flou de Mamdani

La forme générale d'une règle d'un système d'inférence flou de type Mamdani à p entrées, $x_1, x_2, \dots, x_p \in X_1 \times X_2 \times \dots \times X_p$, et une sortie, $y \in Y$, et contenant M règles, est la suivante :

$$R_l : \text{SI } x_1 \text{ est } F_1^l \text{ ET } \dots \text{ ET } x_p \text{ est } F_p^l \text{ ALORS } y \text{ est } G^l,$$

où F_i^l ($i = 1, \dots, p$) sont les ensembles flous (valeurs linguistiques) des prémisses, qui sont représentés par leurs fonctions d'appartenance, $\mu_{F_i^l}$, G^l est l'ensemble flou de la conclusion, représenté par sa fonction d'appartenance μ_{G^l} ($l = 1, \dots, M$).

Une conséquence importante du fait que les règles d'une base sont disjonctives est que, si présent dans la prémisse d'une règle, le connecteur logique de disjonction "OU" peut être éliminé en remplaçant la règle en question avec des nouvelles règles, une pour chaque disjunctif de la prémisse, toutes avec le même conséquent (partie ALORS). Par exemple,

R : SI *truc* OU *muche* ALORS *machin*

est équivalente à

R' : SI *truc* ALORS *machin*

R'' : SI *muche* ALORS *machin*.

Une base de règles où les prémisses ne contiennent pas de connecteurs "OU" est dite en forme normale.

3 Consignes

Soit un système flou conçu pour l'évaluation d'un rapport de stage, qui prend en entrée trois variables,

- Résultats obtenus (sur 20),
- Méthodes utilisées (sur 20),
- Présentation (sur 20),

et renvoie l'évaluation globale du rapport (sur 20).

Pour chaque variable d'entrée, on va définir trois valeurs linguistiques : Médiocre, Moyen, Excellent.

Pour l'évaluation globale, par contre, on va définir cinq valeurs linguistiques : Médiocre, Mauvais, Moyen, Bon, Excellent.

La base de règle sera constituée des six règles suivantes :

R_1 : Si Résultat est Moyen ET Méthodes est Médiocre ALORS Evaluation est Mauvais

R_2 : : Si Résultat est Moyen ET Méthodes est Excellent ALORS Evaluation est Bon

R_3 : : Si Résultat est Médiocre ET Méthodes est Moyen ALORS Evaluation est Mauvais

R_4 : : Si Résultat est Excellent ET Méthodes est Excellent ET Présentation est Excellent
ALORS Evaluation est Excellent

R_5 : : Si Résultat est Médiocre OU Méthodes est Moyen ALORS Evaluation est Moyen

R_6 : : Si Résultat est Moyen OU Méthodes est Médiocre ALORS Evaluation est Médiocre

1. Définissez les fonctions d'appartenance de valeurs linguistiques pour chaque variable.
2. Transformez la base de règles en forme normale.
3. Réalisez ce système en Python en utilisant la bibliothèque `skfuzzy`.
4. Soit un rapport de stage avec les notes :
 - Résultats obtenus : 12,
 - Méthodes utilisées : 6,
 - Présentation : 19 ;appliquez le système pour obtenir l'évaluation globale correspondante.
5. Que dites-vous de cette base de règles ? Est-elle bien établie ?
6. Essayez des définitions alternatives des valeurs linguistique ; changez la méthode de dé-fuzzification. Que'est-ce que vous pouvez constater ?

Rendez votre code, vos réponses aux questions et vos observations dans un archive zippé par courriel.