



WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ

**Sztuczna inteligencja
Laboratorium**

Synteza układu wnioskującego

Stanislau Antanovich
nr. indeksu: 173590
gr. lab: L04

Spis treści

Spis rysunków

1 Wstęp

1.1 Cel ćwiczenia

Laboratorium składa się z trzech zasadniczych części. Część 2 ma na celu zapoznanie się ze sposobem syntezy rozmytego systemu ekspertowego typu Mamdaniego z wykorzystaniem biblioteki **scikit-fuzzy**. W części ??, należy zapoznać się z ideą działania systemu Mamdaniego a następnie dokonać modyfikacji systemu wykonanego w części 2. Część ?? laboratorium polega na wykonaniu przykładowego zadania zaliczeniowego.

2 System ekspertowy typu Mamdaniego

2.1 Problem

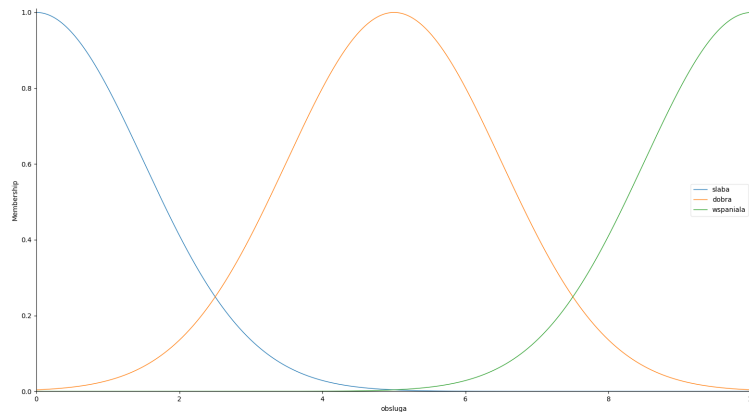
Zaprojektować rozmyty układ ekspertowy doradzający ile napiwku pozostawić w restauracji na podstawie oceny jakości obsługi oraz jakości jedzenia. Jakość obsługi i jakość jedzenia będzie oceniana w skali od 1 do 10, gdzie 10 reprezentuje ocenę maksymalną, natomiast napiwek będzie liczbą z przedziału [0,30] reprezentującą procent wartości rachunku.

Baza reguł będzie składała się z 5 reguł. System zostanie wykonany w dwóch etapach. W **etapie pierwszym**(rys. 1 i ??) system będzie zbudowany z jednego wejścia(*obsługa*) i jednego wyjścia(*napiwek*) oraz 3 reguł postaci:

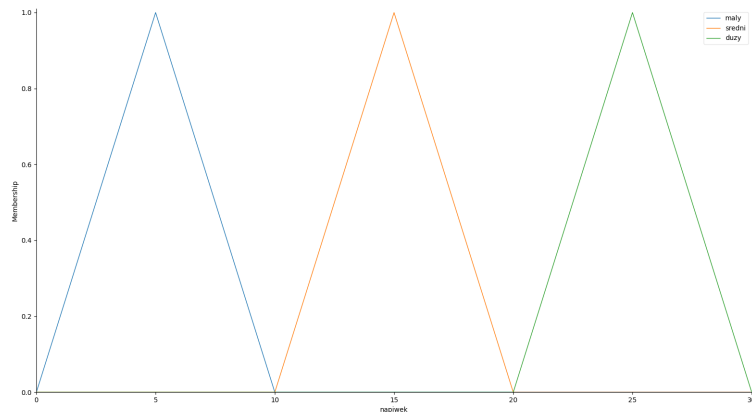
R1 *jeżeli obsługa jest słaba, to napiwek jest mały*

R2 *jeżeli obsługa jest dobra, to napiwek jest średni*

R3 *jeżeli obsługa jest wspaniała, to napiwek jest duży*



Rysunek 1: *Obsługa*



Rysunek 2: *Napiwek*

W **etapie drugim** do systemu zostanie dodane gruga zmienna wejściowa *jedzenie* (rys. ??) oraz 2 dodatkowe reguły

R4 *jeżeli jedzenie jest zepsute, to napiwek jest mały*

R5 *jeżeli jedzenie jest wyborne, to napiwek jest duży*

2.2 Realizacja

2.2.1 Inicjowanie modułów

W tej sekcji inicjujemy wszystkie wymagane biblioteki dla prawidłowego działania programu.

```
1 import numpy as np
2 import skfuzzy as fuzz
3 from skfuzzy import control as ctrl
4 import matplotlib.pyplot as plt
```

2.2.2 Tworzenie zmiennych stanu poprzednika “obsługa” oraz następnika “napiwek”

W tej sekcji tworzymy zmienne stanu poprzednika “obsługa” oraz następnika “napiwek”.

```
1 obsluga = ctrl.Antecedent(np.arange(0, 10.01, 0.01), 'obsługa')
2 napiwek = ctrl.Consequent(np.arange(0, 30.01, 0.01), 'napiwek')
```

2.2.3 Dodanie zbiorów rozmytych

W tej sekcji do zmiennej *obsługa* dodajemy następujące zbiory: *slaba*, *dobra*, *wspaniala*.

Zbiór *slaba* o centrum umieszczonym w punkcie uniwersum równym **0** i rozpiętości wynoszącej **1.5**. Dla zbiorów *dobra* i *wspaniala* o centrach ułożonych w punktach odpowiednio **5** oraz **10** i rozpiętości wynoszącej **1.5**

Dla zmiennej *napiwek* dodajemy zbiory trójkątne: *maly*, *sredni*, *duzy* o parametrach [0, 5, 10], [10, 15, 20] i [20, 25, 30] odpowiednio.

```
1 obsluga['slaba'] = fuzz.gaussmf(obsluga.universe, 0, 1.5)
2 obsluga['dobra'] = fuzz.gaussmf(obsluga.universe, 5, 1.5)
3 obsluga['wspaniala'] = fuzz.gaussmf(obsluga.universe, 10, 1.5)
4
5 napiwek['maly'] = fuzz.trimf(napiwek.universe, [0, 5, 10])
6 napiwek['sredni'] = fuzz.trimf(napiwek.universe, [10, 15, 20])
7 napiwek['duzy'] = fuzz.trimf(napiwek.universe, [20, 25, 30])
```

2.2.4 Podgląd zbiorów rozmytych

2.2.5 Definicja reguł

2.2.6 Dodanie reguł do systemu rozmytego

2.2.7 Sprawdzenie działania systemu

2.2.8 Sprawdzenie działania systemu dla wartości obsługi

2.2.9 Dodanie drugiej wejściowej

2.2.10 Dodanie reguł 4 i 5

2.2.11 Sprawdzenie działania systemu dla wartości “obsługi”

2.2.12 Sprawdzenie działania systemu dla wartości “obsługi” i “jedzenia”

3 Modyfikacja systemu

4 Przykładowe zadania zaliczeniowe

5 Wnioski