Tworzenie rozmytego systemu wnioskowania

Wstęp

W odróżnieniu od klasycznych systemów regałowych modele rozmyte pozwalają budować modele wnioskujące oparte o język naturalny, dzieki czemu inżynierom wiedzy łatwiej jest przenieść naturalną ludzka wiedzę (często niepewna i niejednoznaczną) do systemów ekspertowych umożliwiających automatyczne wnioskowanie.

Poniżej przedstawiono przykład wykorzystania Fuzzy Logic Toolbox do budowy w/w rozmytego systemu wnioskowania.

Budowa systemu FIS (Fuzzy Inference System – rozmyty system wnioskowania)

Lista poleceń j. matlab służących budowie rozmytego systemu wnioskowania:

- Fis = newfis('nazwa') – tworzy nowy rozmyty system wnioskowania o nazwie *'nazwa'*

  fis = newfis('Smak jabłka');

  fis jest w rzeczywistości strukturą o postaci:

  name: 'przykład' – nazwa system rozmytego

  type: 'mamdani' – typ modelu wnioskowania (*mamdani/sugeno*)

  andMethod: 'min' – sposób obliczania operacji 'I'

  orMethod: 'max' – sposób obliczania operacji 'lub'

  defuzzMethod: 'centroid' – sposób realizacji operacji defazyfikacji

  impMethod: 'min' – sposób realizacji operatora implikacji

  aggMethod: 'max' – sposób realizacji operatora agregacji

  input: [1x2 struct] – zbiór zmiennych wejściowych

  output: [1x1 struct] – zbiór zmiennych wyjściowych

  rule: [] – zbiór reguł

- fis = addvar(fis,typ,nazwa,zakres) – dodaje do rozmytego systemu wnioskowania fis nową zmienną lingwistyczną typu *typ* ('input','output'), która przyjmuje zakres zmienności *zakres*
  *P*rzykład:

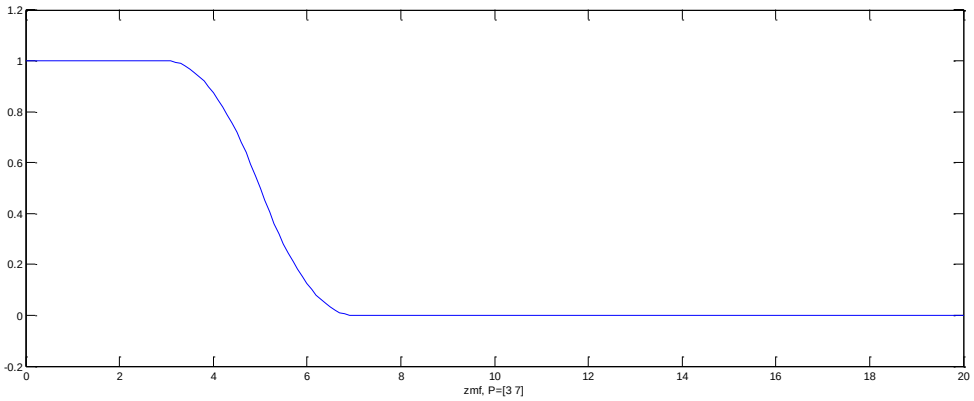
  fis = addvar(fis,'input','kolor',[0 255]);

  fis = addvar(fis,'input','Średnica',[0 20]);

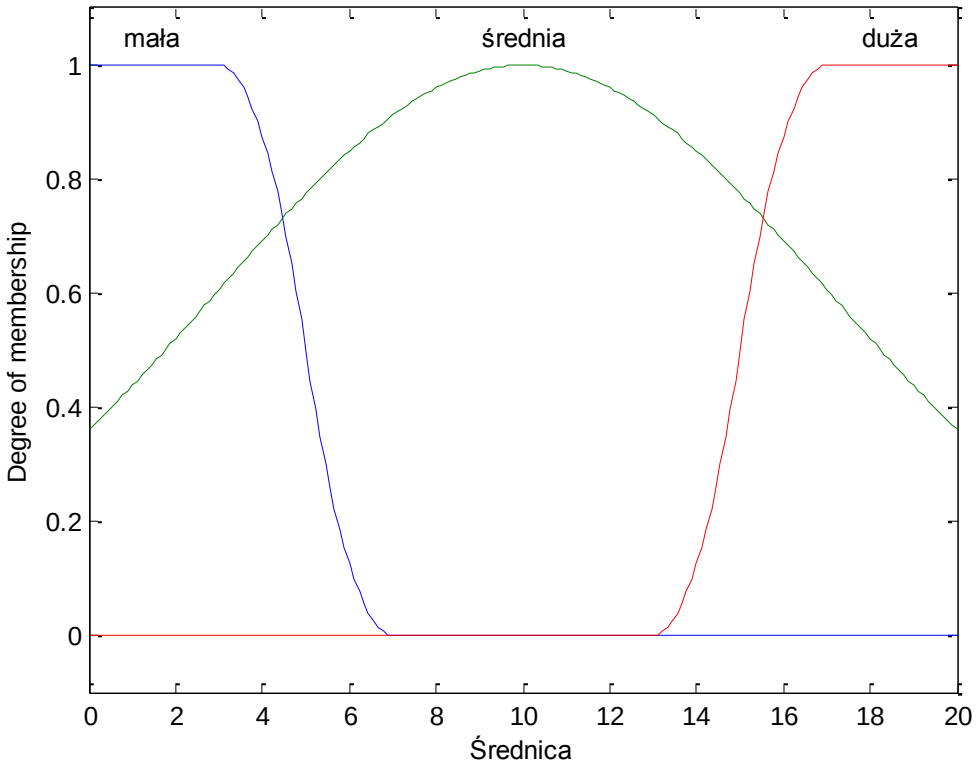
fis = addvar(fis,’output’,’smak’,[0 1]);

- fis = addmf(fis,typ,nr,nazwa_wartosci,typ_mf,param) – dodaje do istniejącej struktury *fis*, do jej wejścia lub wyjścia *typ* = (‘input’/’output’) , o odpowiednim numerze *nr* nowej funkcji przynależności czyli wartości rozmytej o nazwie *nazwa* , która ma kształt *typ_mf* i parametry *param.* Możliwe typy funkcji przynależności:
  ‘gaussmf’,’trimf’,’bellmf’,’trapmf’,’sigmf’,’smf’,’zfm’. Każdy z tych typów ma swój własny zestaw parametrów np.dla *gaussmf* , *param* przyjmuje wartości [a b], gdzie *a*- położenie funkcji Gaussa, *b*-szerokość funkcji Gaussa.
  Przykład:

  fis = addmf(fis,’input’,2,’mała’,’zmf’,[3 7]); %dodaje do wejścia drugiego czyli średnica wartość lingwistyczna *mała*, która ma kształt jak na rysunku:



  W podobny sposób można dodać kolejne wartości lingwistyczne:
  fis = addmf(fis,’input’,2,’średnia’,’gaussmf’,[10 7]);
  fis = addmf(fis,’input’,2,’duża’,’smf’,[13 17]);
  fis = addf(fis,’output’,1,’smaczne’,’smf’,[0.4 0.8]);
  fis = addf(fis,’output’,1,’niedobre’,’zmf’,[0.4 0.8]);



- plotmf(fis,typ,nr) – rysuje wykres funkcji przynależności struktury *fis* powiązanych z odpowiednim numerem zmiennej danego typu *('input','output')*

  Przykład

  Plotmf(fis,'input',2) – system powinien narysować rysunek jak wyżej
- fis = addrule(fis,[m n a b]) – dodaje do struktury *fis* nowej reguły o parametrach:
  m – lista funkcji przynależności związana z danym wejściem w naszym przypadku [0 2] – oznacza to że dla zmiennej 1 (kolor) nie korzystamy z żadnej funkcj przynalezności, natomiast dla drugiej zmiennej (średnica) wybieramy 2 funkcje przynależności „średnia"
  n – lista funkcji przynależności z związana z danym wyjściem (ponieważ mamy tylko jedno wyjście to podajemy nr funkcji przynależności n=2
  a – wartość wagi naszej reguły (na ile ufamy naszej regule) a=1
  b – typ operacji zachodzących między przesłankami 1 -> AND, 2->OR
  Ogólnie więc drugi parametr funkcji addrule ma m+n+2 parametrów

  Przykład

  fis = addrule(fis,[0 2 2 1 1]);

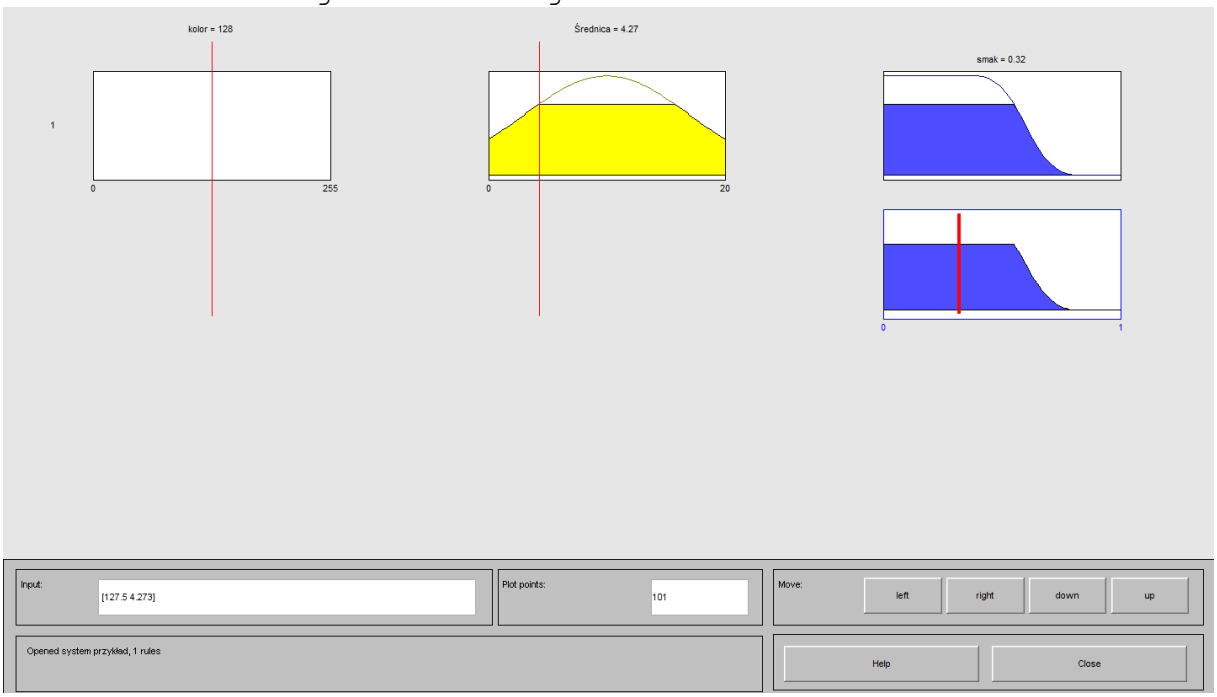
  powoduje dodanie reguły:

  If (Średnica is średnia) then (smak is niedobre) (1)

- showrule(fis) – funkcja wyświetla listę utworzonych reguł (patrz wyżej)
- wyn = evalfis(data,fis) – uruchamia zbudowany system *fis* dla danych *data* (uwaga dane musza mieć odpowiednią ilość zmiennych, czyli taką jaką zdefiniowano podczas definicji zmiennych typu input). Wyn – to wynik predykcji dla danych *data*
  Przykład:

evalfis([45 5],fis)
zwróci wartość 0.3165 oznaczającą że smak przyjmuje wartość około 0.3 czyli dane jabłko jest raczej nie zmaczne

- ruleview(fis) – funkcja umożliwia graficzną analizę zbudowanego zbioru reguł:



- gensurf(fis) – rysuje kształt funkcji decyzyjnej
- fis = setfis(fis,'fispropname','newfisprop') umożliwia modyfikację wybranej właściwości struktury fis
  Przykład
  fis = setfis(fis,'impMethod','prod')
- fis = parserule(fis,'textrule') – umozliwia parsowanie reguły zapisanej jako tekst
  Przykład:

Zadanie

Zad 1.

Stwórz własny rozmyty system wnioskowania umożliwiający wnioskowanie na temat płci na podstawie rozmiaru stopy oraz wzrostu człowieka.

Założenia –

- duże osoby o dużym rozmiarze stopy to mężczyźni
- małe osoby o małym rozmiarze stopy to kobiety

W budowanym systemie zaproponuj przedział zmienności danych zmiennych wejściowych (wzrost, rozmiar stopy), zdefiniuj odpowiednie wartości lingwistyczne dla tych zmiennych oraz określ dla nich funkcje przynależności, następnie utwórz odpowiednie reguły umożliwiające wnioskowanie na temat płci.

Zad 2.

Na powyższym przykładzie zbadaj działanie różnych operatorów koniunkcji. Możliwe wartość jakie może przyjąć operator koniunkcji Fuzzy Logic Toolbox znajdź w Helpie

Zad 3.

Na powyższym przykładzie zbadaj działanie różnych operatorów alternatywy. Możliwe wartość jakie może przyjąć operator koniunkcji Fuzzy Logic Toolbox znajdź w Helpie

Zad 4.

Na powyższym przykładzie zbadaj działanie różnych operatorów implikacji. Możliwe wartość to 'min' oraz 'prod'

Zad 5.

Na powyższym przykładzie zbadaj działanie różnych operatorów agregacji. Możliwe wartość to :

- max – maksimum spośród wartości do agregacji
- sum – suma agregowanych wartości
- pro bor – dla dwóch argumentów funkcji probor *a* i *b* wynikiem jest y = a + b - ab

Zad 6.

Na powyższym przykładzie zbadaj działanie różnych operatorów wyostrzania. Możliwe wartość to:

- centroid – środek obszaru
- bisector- bisekcja obszaru (podział obszaru za pomocą prostej na dwa o równej powierzchni)
- mom – wartość średnia maksimum
- som - najmniejsza wartość maximum
- lom – największa wartość maksimum

UWAGA

Do realizacji powyższych zadań porównania najlepiej jest wykorzystać funkcje *gensurf* oraz *ruleview*