Inteligencja obliczeniowa - Logika rozmyta

Grzegorz Madejski

2021/22

Logika rozmyta

- Klasyczna logika dopuszcza dwie wartości: 0 - fałsz, 1 - prawda.
- W życiu czasem ciężko zdecydować, czy coś jest jednoznacznie prawdziwe, czy fałszywe.

Logika rozmyta

- Klasyczna logika dopuszcza dwie wartości: 0 - fałsz, 1 - prawda.
- W życiu czasem ciężko zdecydować, czy coś jest jednoznacznie prawdziwe, czy fałszywe.
- Przykład: "Obiekt na zdjęciu to samochód". Prawda czy fałsz?

Logika rozmyta

- Klasyczna logika dopuszcza dwie wartości: 0 - fałsz, 1 - prawda.
- W życiu czasem ciężko zdecydować, czy coś jest jednoznacznie prawdziwe, czy fałszywe.
- Przykład: "Obiekt na zdjęciu to samochód". Prawda czy fałsz?

Logika rozmyta

- Klasyczna logika dopuszcza dwie wartości: 0 - fałsz, 1 - prawda.
- W życiu czasem ciężko zdecydować, czy coś jest jednoznacznie prawdziwe, czy fałszywe.
- Przykład: "Obiekt na zdjęciu to samochód". Prawda czy fałsz?

Logika rozmyta

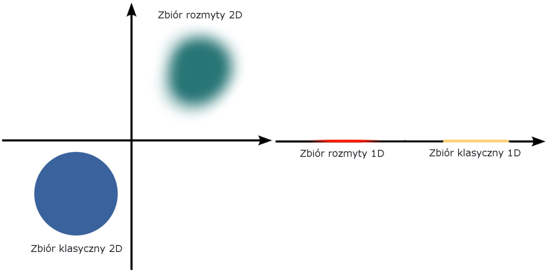
- Zamiast wartości binarnych $\{0, 1\}$, można rozpatrywać wszystkie liczby z przedziału $[0, 1]$.
- Mówimy wówczas o *logice rozmytej* (ang. fuzzy logic).
- Logikę rozmytą (i zbiory rozmyte) stworzył w latach 60. i 70. Lotfi Zadeh.

Zbiory rozmyte

- Logika rozmyta często stosowana jest dla zbiorów. Mówimy wówczas o *zbiorach rozmytych* (ang. fuzzy sets).
- Logika klasyczna (boolowska): element jest w zbiorze, elementu nie ma w zbiorze.
- Logika rozmyta: element jest w zbiorze z pewnym stopniem przynależności (od 0 do 1).

Zbiory rozmyte

- Logika rozmyta często stosowana jest dla zbiorów. Mówimy wówczas o *zbiorach rozmytych* (ang. fuzzy sets).
- Logika klasyczna (boolowska): element jest w zbiorze, elementu nie ma w zbiorze.
- Logika rozmyta: element jest w zbiorze z pewnym stopniem przynależności (od 0 do 1).



Zbiory rozmyte

- Będziemy się zajmować zbiorami w przestrzeni jednowymiarowej (choć wszystkie pojęcia można uogólnić na przestrzeń n-wymiarową).
- Zbiór rozmyty $A \subseteq X$ najlepiej reprezentować za pomocą *funkcji przynależności*:

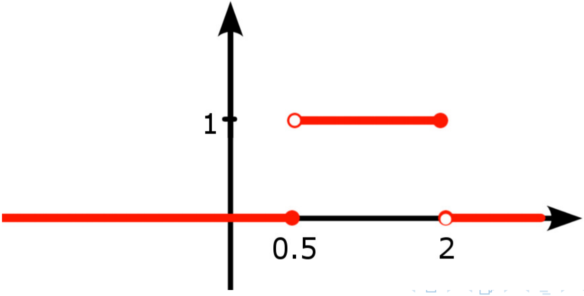
$$\mu_A(x) : X \to [0, 1]$$

Zbiory rozmyte

- W przypadku zbiorów klasycznych (nierozmytych), funkcja ta ma zawsze taki sam wzór:

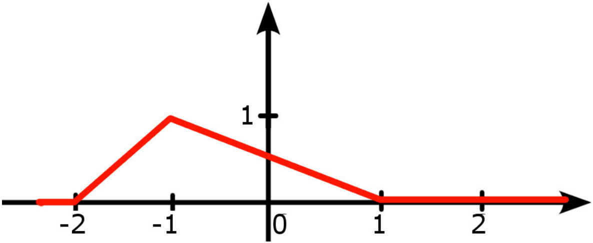
$$\mu_A(x) = \begin{cases} 0, & x \notin A \\ 1, & x \in A \end{cases}$$

- Przykład: zbiór $A = (0.5, 2]$ jest prezentowany przez funkcję:



Zbiory rozmyte

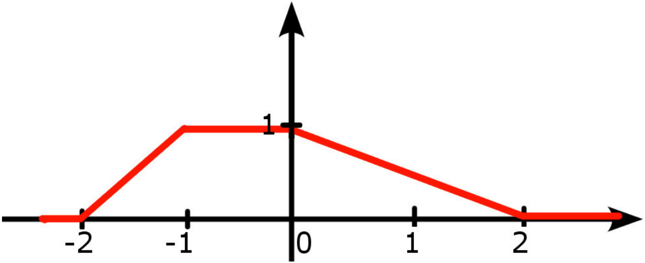
- W przypadku zbiorów rozmytych, funkcja $\mu_A(x) : X \to [0, 1]$ może mieć różne wzory, które inaczej będą określały rozmycie zbioru.
- Funkcja przynależności *trójkątna* (ang. triangular, cone).



$$\mu_A(x) = \begin{cases} 0, & x \leqslant -2 \\ x + 2, & x \in (-2, -1] \\ -0.5 \cdot x + 0.5, & x \in (-1, 1] \\ 0, & x > 1 \end{cases}$$

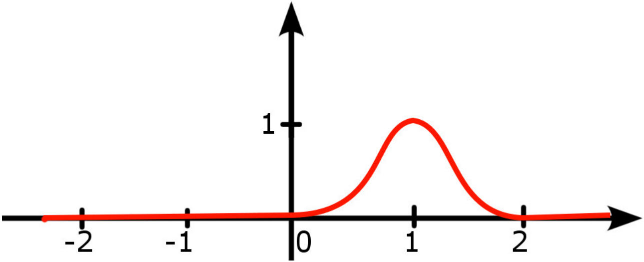
Zbiory rozmyte

- W przypadku zbiorów rozmytych, funkcja $\mu_A(x) : X \rightarrow [0, 1]$ może mieć różne wzory, które inaczej będą określały rozmycie zbioru.
- Funkcja przynależności *trapezoidowa* (ang. trapezoid).



Zbiory rozmyte

- W przypadku zbiorów rozmytych, funkcja $\mu_A(x) : X \rightarrow [0, 1]$ może mieć różne wzory, które inaczej będą określały rozmycie zbioru.
- Funkcja przynależności *gaussowska* (ang. gaussian).



Zbiory rozmyte

- Formalnie, zbiór rozmyty można zdefiniować poprzez jego funkcję przynależności jako:

$$A = \{(x, y) : y = \mu_A(x)\}$$

- Z pojęciem zbioru rozmytego związanych jest kilka pojęć pomocniczych.

Zbiory rozmyte

- *Nośnik* (ang. support) zbioru A to zbiór elementów z A, dla których wartość funkcji przynależności jest większa od 0.

$$Support(A) = \{x : \mu_A(x) > 0\}$$

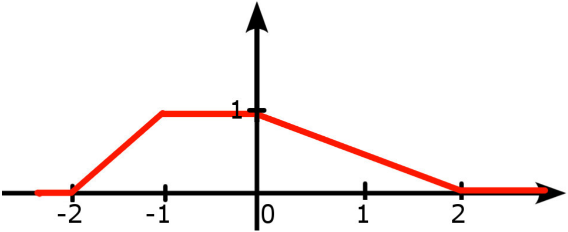
- *Rdzeń* (ang. core) zbioru A to zbiór elementów z A, dla których wartość funkcji przynależności wynosi 1.

$$Core(A) = \{x : \mu_A(x) = 1\}$$

- *Wysokość* (ang. height) zbioru A to największa wartość funkcji przynależności

$$h(A) = \max\{\mu_A(x) : x \in X\}$$

Zbiory rozmyte



- *Nośnik*:

$$Support(A) = \{x : \mu_A(x) > 0\} = (-2, 2)$$

- *Rdzeń*:

$$Core(A) = \{x : \mu_A(x) = 1\} = [-1, 0]$$

- *Wysokość*:

$$h(A) = \max\{\mu_A(x) : x \in X\} = 1$$

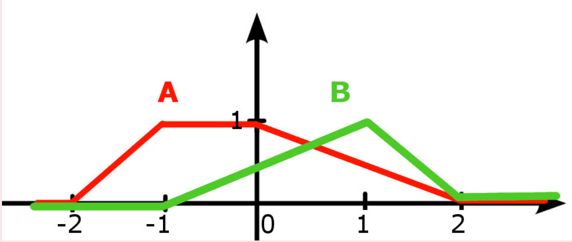
Operacje na zbiorach rozmytych

- *Dopełnienie* zbioru rozmytego A to zbiór A^c (lub $\bar{A}$), którego funkcja przynależności dana jest wzorem $\mu_{A^c}(x) = 1 - \mu_A(x)$.
- *Suma* zbiorów rozmytych A i B to zbiór $A \cup B$, którego funkcja przynależności dana jest wzorem $\mu_{A \cup B}(x) = s(\mu_A(x), \mu_B(x))$, gdzie s to pewna funkcja zwana s-normą.
- Powszechnie stosowaną s-normą jest funkcja max.
- *Przekrój* (cześć wspólna) zbiorów rozmytych A i B to zbiór $A \cap B$, którego funkcja przynależności dana jest wzorem $\mu_{A \cap B}(x) = t(\mu_A(x), \mu_B(x))$, gdzie t to pewna funkcja zwana t-normą.
- Powszechnie stosowaną t-normą jest funkcja min.

Operacje na zbiorach rozmytych

Zadanie

Dane są dwa zbiory rozmyte A i B:

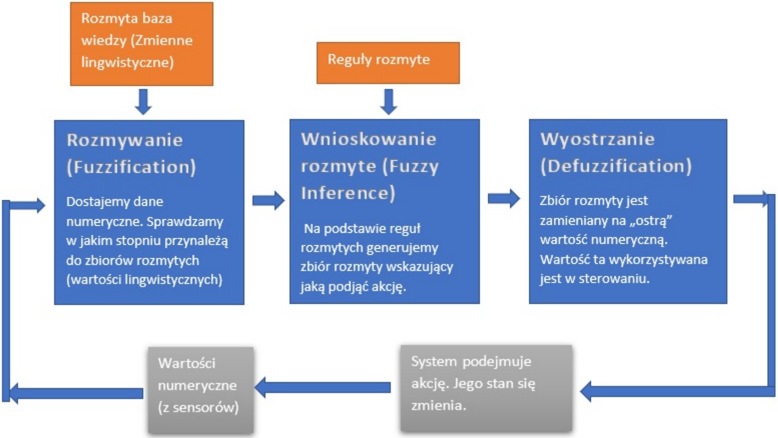


- Narysuj na osobnych wykresach funkcje przynależności dla zbioru $A \cup B$ (wykorzystując s-normę max) i funkcję przynależności dla zbioru $A \cap B$ (wykorzystując t-normę min).
- Dla obu obliczonych zbiorów podaj nośnik, rdzeń i wysokość.

Kontroler rozmyty

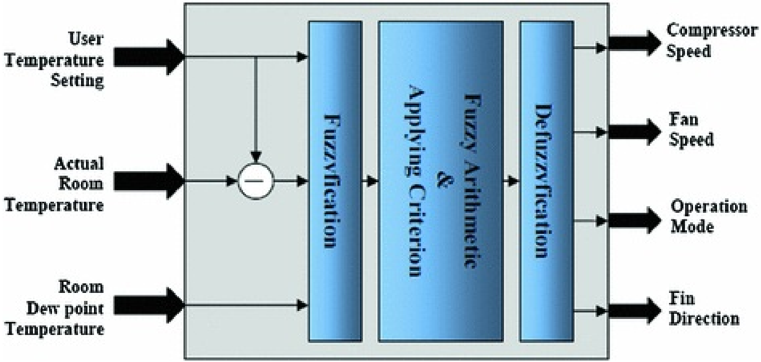
- Jak wykorzystać zbiory rozmyte w praktyce?
- *Kontroler rozmyty* (ang. Fuzzy Control System, Fuzzy Controller) to system sterujący, który pobiera dane numeryczne i podejmuje decyzje o swoich akcjach, wykorzystując logikę i zbiory rozmyte.
- Kontrolery rozmyte mogą: sterować klimatyzacją, sterować sprzętem AGD (np. lodówka), sterować autonomicznymi autami (hamowanie, parkowanie), podejmować trudne decyzje (np. udzielanie kredytu, szacowanie ryzyka).

Kontroler rozmyty



Kontroler rozmyty

Przykład: klimatyzacja.



Zmienne lingwistyczne

- Aby kontroler rozmyty działał, trzeba zdefiniować w nim dwa komponenty *zmienne lingwistyczne* (baza wiedzy) i *reguły rozmyte* (baza reguł).
- Dobra konstrukcja obu tych komponentów nie jest łatwa. Wymaga testowania (metoda prób i błędów) oraz wiedzy specjalistycznej i ostrożnego projektowania.

Zmienne lingwistyczne

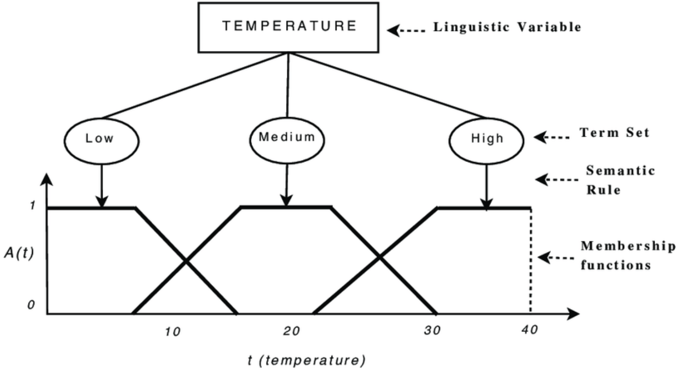
- Aby kontroler rozmyty działał, trzeba zdefiniować w nim dwa komponenty *zmienne lingwistyczne* (baza wiedzy) i *reguły rozmyte* (baza reguł).
- Dobra konstrukcja obu tych komponentów nie jest łatwa. Wymaga testowania (metoda prób i błędów) oraz wiedzy specjalistycznej i ostrożnego projektowania.

Zmienne lingwistyczne

- W systemie sterującym mamy do czynienia z wartościami liczbowymi, które nazywamy *zmienną bazową*.
- Dla zmiennej bazowej, można zdefiniować *zmienną lingwistyczną*, która będzie zestawem zbiorów rozmytych, zwanych *wartościami lingwistycznymi*.
- Zbiory te muszą mieć sensowną etykietkę tekstową (stąd: "lingwistyczne"!). Znaczenie etykiety musi być dobrane sensownym sposobem (sematic rule).

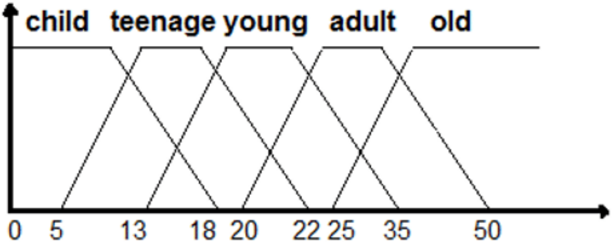
Zmienne lingwistyczne

Przykład: zmienna lingwistyczna "Temperatura" z trzema wartościami lingwistycznymi "Niska", "Średnia", "Wysoka". Wartości lingwistyczne to zbiory rozmyte o trapezoidalnej funkcji przynależności.



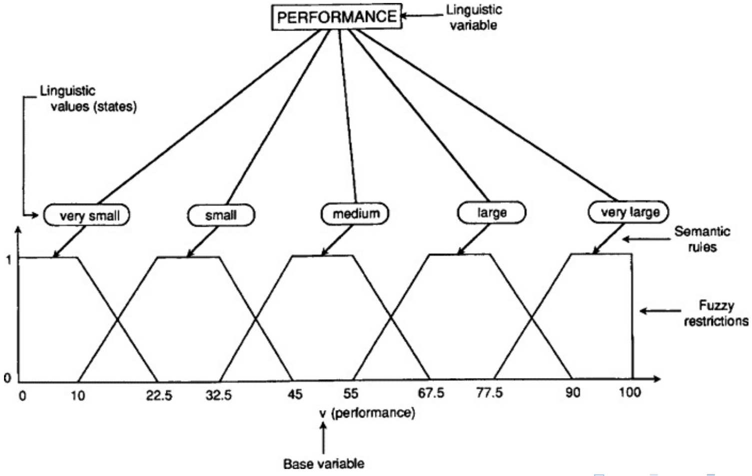
Zmienne lingwistyczne

Przykład: zmienna lingwistyczna "Wiek" z pięcioma wartościami.



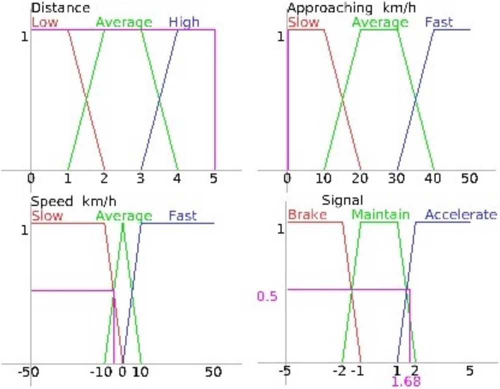
Zmienne lingwistyczne

Przykład: zmienna lingwistyczna "Performance".



Zmienne lingwistyczne

W kontrolerze rozmytym wszystkim zmiennym bazowym należy zaprojektować zmienne lingwistyczne z sensownym wartościami i funkcjami przynależności.



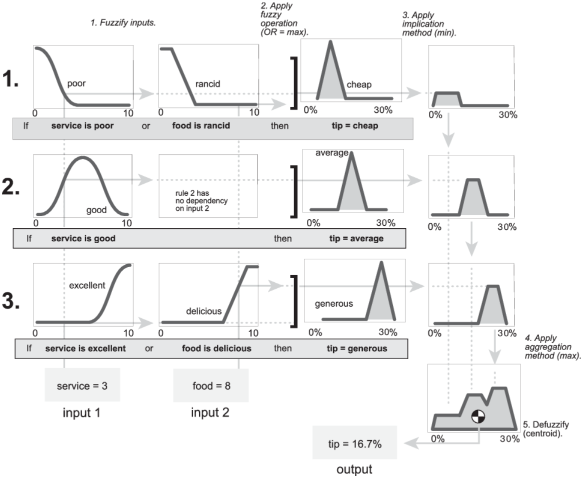
Reguły

- Oprócz zmiennych lingwistycznych należy w naszym kontrolerze umieścić zestaw reguł, które posłużą nam do wnioskowania.
- Reguły mają postać implikacji. Po lewej stronie zapisujemy w alternatywie lub koniunkcji odczyty wejścia (wykorzystując wartości lingwistyczne!), po prawej wskazujemy co robić ze zmienną sterującą.
- Przykład:
  $Temp \in High \wedge Humidity \in Medium \Rightarrow FanSpeed \in High.$

Reguły

- Załóżmy, że chcemy zrobić prosty system dawania napiwków.
- Zmienne lingwistyczne to "service" (obsługa) i "food" (jedzenie), oraz zmienna do sterowania "tip" (napiwek). Każdą z nich zdefiniowaliśmy jako zestaw zbiorów rozmytych.
- Możemy stworzyć trzy reguły "Jeśli obsługa jest słaba lub jedzenie nieświeże, to napiwek mały", "Jeśli obsługa jest dobra, to napiwek średni", "Jeśli obsługa wspaniała lub jedzenie pyszne, to napiwek duży".
- Jak stosujemy te reguły w praktyce? Wnioskowanie rozmyte.

Reguły


1. Fuzzify inputs.
2. Apply fuzzy operation (OR = max).
3. Apply implication method (min).
1.
poor
0
10
rancid
0
10
cheap
0%
30%
If service is poor or food is rancid then tip = cheap
2.
good
rule 2 has no dependency on input 2
average
If service is good then tip = average
3.
excellent
delicious
generous
If service is excellent or food is delicious then tip = generous
service = 3
input 1
food = 8
input 2
4. Apply aggregation method (max).
5. Defuzzify (centroid).
tip = 16.7%
output

Reguły

- Klienci oceniają obsługę na 3 i jedzenie na 8.
- Następuje rozmycie inputów. Rozpatrujemy lewe strony wszystkich trzech reguł. Nakładamy $service = 3$ i $food = 8$ na osie i sprawdzamy ich przynależność do zbioru rozmytego uwzględnionego w regule.
- Mamy dwa poziomy przynależności dla każdej reguły (poza drugą). Ponieważ kwantyfikatory to "lub" to wybieramy większy z nich. W przypadku "i" należy wybierać mniejszy.
- Poziom ten nakładamy wartość lingwistyczną z prawej strony reguły ścinając górę wykresu. Działa tutaj funkcja min (ten sposób nazywamy sterownikiem Mamdaniego).
- Alternatywnie można użyć funkcji mnożącej (sterownik Larsena).

Reguły

Inne fajne przykłady:

- Kontroler rozmyty: sterowanie robotem. https://codecrucks.com/mamdani-fuzzy-inference-method-example/
- Kontroler rozmyty: sterowanie klimatyzacją/ogrzewaniem. https://en.wikipedia.org/wiki/Fuzzy_control_system#Fuzzy_sets

Wyostrzanie

- Dostając zbiór rozmyty po zastosowaniu reguł, chcemy zamienić go na liczbę.
- Proces ten nazywamy *wyostrzaniem* (defuzzification).
- Można zrobić to poprzez wyznaczenie punktu ciężkości wykresu (centroid).
- Na poprzednim rysunku z wykresu po wyostrzeniu wyszło: $tip = 16,7\%$.