# Metody numeryczne w elektrotechnice

Ćwiczenie 14 — Systemy wnioskowania rozmytego

Dariusz Borkowski

## Wstęp

Zrealizujesz przykładowe systemy wnioskowania rozmytego stosowane jako systemy ekspertowe lub w układach sterowania.

## 1 System ekspertowy przewidujący ryzyko wypadku — 3 pkt

Rozważmy problem wyznaczania prawdopodobieństwa wypadku drogowego (czyli ryzyka r) w zależności od prędkości p pojazdu i widzialności w. Jest to problem, który można opisać funkcją dwóch zmiennych. Najprostszym podejściem będzie przyjęcie funkcji liniowej:

$$r = f(p, w) = ap + bw + c$$

(płaszczyzna w przestrzeni 3D) oraz jej współczynników.

Przyjmijmy, że rozpatrujemy przedział prędkości od 0 do 200 km/h oraz widzialność w zakresie od 0 do 3 km. Aby określić współczynniki funkcji liniowej dwóch zmiennych (płaszczyzny) wystarczą 3 punkty. Weźmy następujące 3 punkty (p, w, r) równe (0, 3, 0) (przy zerowej prędkości prawdopodobieństwo wypadku jest zerowe), (100, 2, 0.45) (średnia prędkość, niezła widoczność, rysyko przeciętne), (200, 0, 0.99) (przypadek w stylu "anielski orszak niech twą duszę przyjmie...").

Mając te trzy punkty możemy wyznaczyć współczynniki a, b i c rozwiązując następujący układ równań liniowych:

$$\begin{bmatrix} p_1 & w_1 & 1 \\ p_2 & w_2 & 1 \\ p_3 & w_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$
(1)

Rozwiązanie dla podanych danych daje funkcję jak na rysunku 1. Jak widać, funkcja ta ma zaletę polegającą na tym, że jest liniowa (gładka, łatwa, itd), ale ma też wagę polegającą na tym, że jest liniowa. Z tego powodu próba zapewnienia zerowego prawdopodobieństwa przy zerowej prędkoścu (co ma sens niezależnie od widoczności) skończy się tym, że dla świetnej widoczności 3 km i prędkości 200 km/h prawdopodobieństwo będzie wynosiło także 99% podczas gdy powinno być nieco mniejsze.



Rysunek 1: Funkcja liniowa do wyznaczania ryzyka wypadku.

Wniosek płynący z tego jest taki, że powinniśmy użyć zależności nieliniowej. Oczywiście można przyjąć kilka punktów, w których określone jest prawdopodobieństwo, przyjąć pewien wzór nieliniowej funkcji (np. wielomian dwóch zmiennych) i poprzez rozwiązanie odpowiedniego układu równań określić współczynniki tej funkcji. Może być to proces kłopotliwy i nie zawsze dający sensowne rezultaty, gdyż przyjęte punk-ty charakterystyczne mogą nie należeć do założonej funkcji co oczywiście spowoduje błędy przybliżenia (aproksymacja zamiast interpolacji). Przy większej ilości zmiennych i złożonych zależnościach pomiędzy wejściami i wyjściem jest to znacznie trudniejsze.

Alternatywą jest zbudowanie systemu wnioskowania rozmytego (FIS — fuzzy inference system), w którym wzór nieliniowej skomplikowanej funkcji zastępujemy lingwistycznymi regułami wnioskowania (np. JEŚLI jedziesz za szybko ORAZ widzialność jest marna WTEDY prawdopodobieństwo wypadku jest wysokie). Odpowiednio zaprojektowany system FIS realizuje nieliniowe przetwarzanie zmiennych wejściowych na wyjściową.

Wykonaj w Matlabie polecenie fuzzy w celu włączenia tzw. FIS editor (rysunek 2). Zaznacz symbol podpisany Input1 i zmień poniżej jego nazwę na "predkość". Dodaj drugą zmienną wejściową za pomocę menu Edit > Add variable > Input i zmień jej nazwę na "widzialność". Zmień nazwę zmiennej wyjściowej na "ryzyko". Kliknij dwukrotnie na symbolu dowolnej zmiennej w celu otwarcia edytora funkcji przynależności. Tam ustaw podane wcześniej zakresy zmiennych wejściowych oraz podziel je na następujące zbiory rozmyte (prędkość: mała, średnia, duża, ogromna oraz widoczność: słaba, przeciętna, dobra) według własnego uznania. Możesz zmieniać kształ funkcji przynależności oraz ich parametry określające m.in. zasięg zbioru. Zmienną wyjściową rysyko podziel na 4 zbiory rozmyte: niskie, średnie, znaczne, wy-sokie. Wróć do edytora FIS i kliknij na białym prostokącie pośrodku okna w celu otwarcia edytora reguł. Przemyśl problem i wprowadź kilka reguł (postaraj się nie wprowadzać ich więcej niż 10) lingwistycznych opisujących prawdopodobieństwo wypadku zależne od prędkości i widoczności.



Rysunek 2: FIS Editor i narzędzia do edycji reguł (Rule Editor) oraz do edycji funkcji przynależności (Membership Function Editor).

Po wprowadzeniu reguł wyeksportuj FIS do pliku oraz do zmiennej w Matlab workspace. Następnie wybierz z menu View pozycje Rules i Surface. Otwarte zostaną okna prezentujące działanie poszczególnych reguł (Rule Viewer) oraz okno pokazujące powierzchnię (Surface Viewer, rys. 3) funkcji 2 zmiennych będącej wynikiem postaci reguł i funkcji przynależności . Przeanalizuj dokładnie wyrysowanę powierzchnię pod kątem znalezienia tam miejsc sprzecznych z Twoją wiedzą o problemie np. obszar w którym ze wzrostem prędkości prawdopodobieństwo wypadku maleje. Używając procesu "tentegowania w głowie" lub na drodze eksperymentalnej wyjaśnij przyczynę problemu.



Rysunek 3: Wynikowa nieliniowa funkcja rozmytego systemu wnioskowania do oceny ryzyka wypadku.

Przetestuj wpływ rodzaju i położenia funkcji przynależności zmiennych wejściowych i wyjściowych na wynik (powierzchnia aktualizuje się automatyczne po zmianie dowolnego parametru lub reguły). W oknie podglądu reguł (Rule Viewer) sprawdź jak na wyniki wpływają zmiany implementacji funkcji logicznych, implikacji, agregacji i ostrzenia (defuzyfikacji) dokonywane w oknie FIS Editor. W oknie Rule Viewer sprawdź jaki jest zakres zmian prawodopodbieństwa wypadku dla skrajnych wartości prędkości i widoczności. Wyjaśnij wynik. Porównaj wynikową powierzchnię z funkcją liniową z rysunku 1. Czy potrafisz zaproponować wzór funkcji dwóch zmiennych którego wykres przypominałby wynikową powierzchnię systemu FIS?

Narysuj na kartce kratkę określającą podział (jak na rysunku 4) obu zmiennych na użyte w zadaniu zbiory (nie rozmyte). Następnie zaznacz w niej liczbami stopień ryzyka wynikający z zadanych reguł (4 – wysokie, 3 – znaczne, 2 – średnie, 1 – małe) oraz numer reguły z której to ryzyko wynika. Jeżeli w danym polu działa jednocześnie więcej niż jedna reguła to wpisz wartości ryzyka i numery reguł z których wynikają. Pola na które nie działa żadne ryzyko zostaw puste. Odpowiedz na pytanie, co stanie się po zaprogramowaniu dwóch sprzecznych reguł (dających różny wynik w tej samej sytuacji) w typowym sterowniku przemysłowym (np. PLC).



Rysunek 4: Tabela z zaznaczonymi zgrubnie obszarami działania reguł.

#### 2 System ANFIS do pozycjonowania ramienia robota — 3 pkt

Chcesz zbudować system sterowania ramieniem robotem złożonym z dwóch szeregowo połączonych segmentów poruszających się w jednej płaszczyźnie (rys. 5). System ten na podstawie zadanego położenia (x, y) końca ramienia będzie obliczał kąt  $\theta_1$  pomiędzy pierwszym segmentem a podstawą oraz kąt  $\theta_2$  pomiędzy dwoma segmentami niezbędne do wysterowania silników krokowych realizujących zmiany kątów segmentów.



Rysunek 5: Schematyczny rysunek dwusegmentowego ramienia robota z zaznaczonymi kątami  $\theta_1$  i  $\theta_2$ .

Do realizacji tego zadania można użyć ANFIS (adaptive neuro-fuzzy inference system). Jest to system wnioskowania rozmytego o strukturze podobnej do sieci neuronowej, który w miejscach neuronów posiada reguły wnioskowania rozmytego. Systemy takie uczy się w trybie supervised (pod nadzorem lub inaczej z nauczycielem) podając na wejścia kolejne ciągi uczące oraz na wyjścia znane, pożądane wartości wyjściowe. Nauka polega na adaptacji parametrów funkcji przynależności (położenie, szerokość, itp.) przy znanej struktrurze wynikającej m.in. z ilości zmiennych wejściowych, z ilości zbiorów opisujących zmienne wejściowe i postaci reguł wnioskowania (AND lub OR). Adaptacja ma na celu minimalizację różnic pomiędzy wyjściem ANFIS a zadaną wartością wyjścia dla różnych sygnałów wejściowych. Do adaptacji nadają się jedynie systemy typu Sugeno. Nie opracowano metod uczenia dla systemów Mamdaniego.



Rysunek 6: Narzędzie do podglądu struktury ANFIS.

Niniejsze zadanie wymaga zaprojektowania i nauczenia dwóch systemów ANFIS obliczania kątów  $\theta_1$ i  $\theta_2$ . Wejściami do obu systemów będą współrzędne kartezjańskie (x, y). Wyjściem pierwszego systemu będzie kąt  $\theta_1$ , a drugiego kąt  $\theta_2$ . Konieczne będzie przygotowanie danych uczących czyli zestawu punktów  $(x, y, \theta_1, \theta_2)$ .

Zestawy takie można wygenerować wprost (tzw. direct kinematic) czyli zadać oba kąty i wyliczyć z zależności trygonometrycznych współrzędne (x, y) końca ramienia. Można też postąpić na odwrót (tzw.

inverse kinematic) czyli użyć równań problemu odwrotnego, które dla zadanego punktu (x, y) obliczają wartości kątów. To drugie zadanie jest znacznie trudniejsze. W zasadzie obojętne, której drogi użyjemy, ale powinniśmy wygenerować możliwie szeroki zakres danych pokrywający najlepiej całą dostępną przestrzeń położeń końca ramienia robota. Braki w danych uczących mogą skutkować zaskakującym działaniem sterownika (co zobaczysz w dalszej części zadania).

Dla uproszczenia założymy, że kąt  $\theta_1$  może się zmieniać w granicach od 0 do  $\frac{\pi}{2}$ , a kąt  $\theta_2$  w granicach od 0 do  $\pi$ . Długości segmentów ramienia to  $l_1 = 10$  i  $l_2 = 7$ . Uzupełnij szkielet programu lab14z02.m w celu wygenerowania ciągu uczącego, wygenerowania i nauczenia dwóch systemów ANFIS a następnie porównania wyników nauki z zadanymi wartościami wyjść z ciągu uczącego. Dane do nauki ANFIS wygeneruj korzystając z poniższych zależności na wartości położenia końca ramienia robota:

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \tag{2}$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \tag{3}$$

Przyjmij, że kąty będą zmieniać się co0.1 radiana.

Uczenie systemu ANFIS (poleceniem anfis) zajmuje trochę czasu (do kilku minut), więc wymagana jest pewna doza cierpliwości. Polecenie anfis wymaga podania m.in. nazwy systemu ANFIS, ilości funkcji przynależności wszystkich wejść i wyjść, ilości epok nauki.

Na końcu, po sprawdzeniu działania systemu sterowania zapoznaj się z edytorem systemów ANFIS wykonując polecnie anfisedit(anfis2). Zobacz stukturę systemu oraz kształt realizowanej funkcji (możliwe dlatego, że mamy problem o tylko dwóch wejściach). Spróbuj dokonać zmian w systemie i zobacz jak to się odbije na kształcie funkcji.

#### 3 Układ sterowania systemem centralnego ogrzewania w Simulinku — 3 pkt