Clasificare substractivă. Modelarea unei funcții neliniare de două variabile pe bază de date numerice

Obiective: înțelegerea conceptului și metodelor de clasificare a datelor, înțelegerea noțiunilor de clasă și centru al clasei, familiarizarea cu modul de determinare a preciziei unui model.

Observație: MATLAB/Simulink se accesează online (https://matlab.mathworks.com/), prin logare cu credențialele MS Teams (cele de tip nume.prenume@student.utcluj.ro).

Termeni și acronime: *clasificare tranșantă, clasificare substractivă, Fuzzy C-Means.*

○ Clasificarea datelor

Clasificarea este o metodă fundamentală de analiză a datelor, ce are ca scop identificarea grupării naturale a datelor dintr-un set de date de dimensiuni mari. Clasificarea datelor este o metodă de învățare nesupervizată, deoarece valoarea dorită a ieșirii (număr de clase, apartenența fiecărui obiect la o anumită clasă) nu este cunoscută a priori. Aplicații tipice ale clasificării sunt: recunoașterea formelor, extragerea de caracteristici, cuantizarea vectorilor, segmentarea imaginilor, aproximarea funcțiilor, data mining.

Împărțirea în clase se realizează pe baza unei mulțimi de caracteristici ce descriu fiecare obiect. Rezultatul clasificării este o structură fixă a partiționării datelor, adică pentru fiecare clasă se va cunoaște:

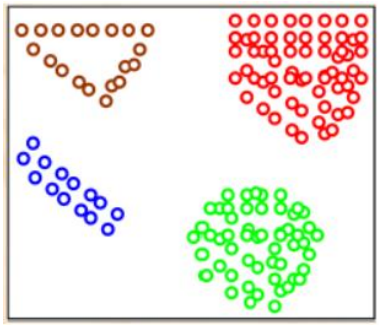
- localizarea - centrul clasei
- forma clasei
- gradul de apartenență al fiecărui obiect la clasa respectivă.

Partiționarea datelor trebuie să respecte:

- omogenitatea în clase - obiectele din aceeași clasă să fie cât se poate de asemănătoare unele cu altele

eterogeneitate între clase - obiectele din clase diferite să fie cât se poate de diferite unele de altele

Un exemplu de măsură a similitudinii dintre obiecte este distanța euclidiană, ilustrată în imaginea de mai jos, unde obiectele care sunt apropiate între ele au fost colorate utilizând aceeași culoare.



Datele de clasificat se reprezintă sub forma vectorilor *N*-dimensionali:

$$X_i = [x_{i1}, x_{i2}, \dots, x_{iN}], X_i \in \mathbb{R}^N, i = \overline{1, \dots, M}$$

unde *N* – numărul caracteristicilor fiecărui obiect, *M* – numărul de obiecte (dimensiunea setului de date).

$$X = \begin{bmatrix} x_{11} & x_{12} \dots & x_{1N} \\ x_{21} & x_{22} \dots & x_{2N} \\ \dots & \dots & \dots \\ x_{M1} & x_{M2} \dots & x_{MN} \end{bmatrix}$$

Obiectivul clasificării este de a găsi cei *K* vectori, care constituie centrele celor *K* clase în care se realizează partiționarea datelor:

$$c_k = [x_{k1}, x_{k2}, \dots, x_{kN}], k = \overline{1, \dots, C}$$

○ Tipuri de clasificare

În ***clasificarea tranșantă***, un obiect aparține în totalitate unei clase, sau nu aparține deloc clasei respective. Astfel, gradul de apartenență al unui obiect la o clasă poate fi 0 sau 1. Fiecare obiect aparține unei clase și nu există clase vide, nici clase care conțin toate obiectele.

În situații reale, clasificarea tranșantă este rareori întâlnită, astfel că se preferă o partiționare a obiectelor în care un obiect să poată aparține mai multor clase în același timp, cu diferite grade de apartenență, cuprinse între 0 și 1. Există două astfel de tipuri de partiționări:

- clasificare fuzzy – Fuzzy C-Means, clasificare substractivă
- clasificare probabilistică.

○ Clasificarea substractivă

Clasificarea substractivă determină numărul de clase și centrele claselor dintr-un set de date. Trebuie precizată valoarea unei raze ce specifică domeniul de influență al centrului clasei, în fiecare dimensiune a datelor.

Etapele algoritmului de clasificare substractivă sunt:

1. Presupune că fiecare punct de date este un potențial centru de clasă și calculează probabilitatea ca acesta să definească un centru, pe baza densității punctelor înconjurătoare.
2. Selectează punctul cu cel mai mare potențial ca fiind primul centru de clasă.
3. Înlătură toate punctele din vecinătatea centrului determinat anterior (în conformitate cu raza precizată), în scopul determinării următoarei clase și a centrului ei.
4. Repetă acest proces până când toate punctele se află în raza de influență a unui centru de clasă.

○ Modelarea unei funcții neliniare de două variabile. Aplicație la modelarea funcțională a unui amplificator transconductanță

Descărcați arhiva "*Date_Machete.zip*" și plasați conținutul acesteia (cu *drag-and-drop*) în directorul curent în care lucrează MATLAB. Dezarhivarea se face cu dublu click pe fișier.

http://www.bel.utcluj.ro/dce/didactic/sf/lab/10ModelareFunctionala/Date_Machete.zip

Arhiva conține 6 fișiere: 3 fișiere de tip *.mat*, în care se regăsesc seturile de date, respectiv 3 script-uri cu extensia *.m*.

Pentru construirea modelului care implementează funcția *amplificare = f(frecvență, temperatură)* se utilizează trei seturi de date:

- "*gendata*" cu dimensiunea 179, pentru generarea sistemului fuzzy inițial
- "*antdata*" cu dimensiunea 35717, pentru antrenarea sistemului fuzzy inițial
- "*verdata*" cu dimensiunea 497, pentru verificarea sistemului fuzzy pe durata antrenării (în scopul detectării suprapotrivirii, dacă este cazul)

Domeniul inițial de variație al mărimilor de intrare este:

frecventa_ini: [1 Hz, 10 MHz]

temperatura_ini: [-55, +125]℃

Pentru a reduce gama dinamică a datelor de intrare și pentru a lucra doar cu valori pozitive, se efectuează următoarele transformări:

frecventa = *log(frecventa_ini)*, astfel frecvența va fi cuprinsă în intervalul [0, 7]

temperatura = *temperatura_ini* + 60, astfel temperatura va fi cuprinsa in intervalul [5, 185]℃

Seturile de date au o structură matricială, fiecare rând al matricii conținând un tuplu de date [*frecvență, temperatură, amplificare*].

Exercițiul 1

Examinați structura celor 3 seturi de date, prin încărcare în workspace, cu dublu-click pe fiecare fișier *.mat*.

Exercițiul 2

Pentru **generarea sistemului fuzzy inițial**, se utilizează script-ul "*generare_aft.m*", în care se va completa cu secvențe de cod, conform instrucțiunilor din fișier.

Generați sistemul fuzzy inițial, utilizând funcția *genfis2*. Examinați proprietățile acestuia, din *Fuzzy Logic Designer*.

Ce valoare are parametrul *radii*?

În câte clase a fost împărțit setul de date?

Câte reguli are sistemul fuzzy generat?

Care este legătura dintre numărul de clase, numărul de reguli și structura fiecărei reguli?

Ce tip au mulțimile fuzzy de intrare? Care este expresia mulțimilor fuzzy de ieșire?

Care sunt centrele claselor, pentru variabilele de intrare? Vizualizați suprafața de control a sistemului fuzzy inițial.

Exercițiul 3

Pentru **determinarea preciziei de aproximare a modelului fuzzy inițial**, se utilizează script-ul "*erori.m*", în care se va completa cu secvențe de cod, conform instrucțiunilor din fișier.

Calculați și salvați într-un fișier:

- valoarea maximă a erorii absolute

$$eroare\_absolută_i = |amplificare\_referință_i - amplificare\_fuzzy_i|$$

- valoarea maximă a erorii relative

$$eroare\_relativă_i = \frac{|amplificare\_referință_i - amplificare\_fuzzy_i|}{amplificare\_referință_i}$$

- eroarea medie procentuală

$$eroare\_medie\_procentuală = \frac{1}{N}\sum_{i=1}^{N}\frac{|amplificare\_referință_i - amplificare\_fuzzy_i|}{amplificare\_referință_i}$$

Exercițiul 4

Pentru **antrenarea sistemului fuzzy inițial**, se utilizează script-ul "*antrenare_aft.m*", în care se va completa cu secvențe de cod, conform instrucțiunilor din fișier.

Antrenați sistemul fuzzy inițial, utilizând funcția *anfis*.

Ce tip de eroare utilizează funcția *anfis*?

Comentați evoluția erorilor pentru seturile de date de antrenare și de verificare. Apare fenomenul de suprapotrivire?

Considerați că ar fi utilă creșterea numărului de epoci de antrenare? Justificați.

Vizualizați suprafața de control a sistemului fuzzy antrenat. Comparați-o cu cea a sistemului fuzzy inițial.

Identificați modificările apărute în mulțimile fuzzy de intrare și de ieșire, pe durata antrenării.

Exercițiul 5

Pentru **determinarea preciziei de aproximare a modelului fuzzy final**, se utilizează script-ul "*erori.m*", în care se va completa cu secvențe de cod, conform instrucțiunilor din fișier.

Calculați și salvați într-un fișier:

- valoarea maximă a erorii absolute

$$eroare\_absolută_i = |amplificare\_referință_i - amplificare\_fuzzy_i|$$

- valoarea maximă a erorii relative

$$eroare\_relativă_i = \frac{|amplificare\_referință_i - amplificare\_fuzzy_i|}{amplificare\_referință_i}$$

- eroarea medie procentuală

$$eroare\_medie\_procentuală = \frac{1}{N}\sum_{i=1}^{N}\frac{|amplificare\_referință_i - amplificare\_fuzzy_i|}{amplificare\_referință_i}$$

Comparați valorile obținute pentru sistemul fuzzy final cu cele obținute pentru sistemul fuzzy inițial.