

## Ćwiczenie KM

# Klasyfikator minimalnoodległościowy

### Część teoretyczna

Wykład na temat klasyfikatorów minimalnoodległościowych.

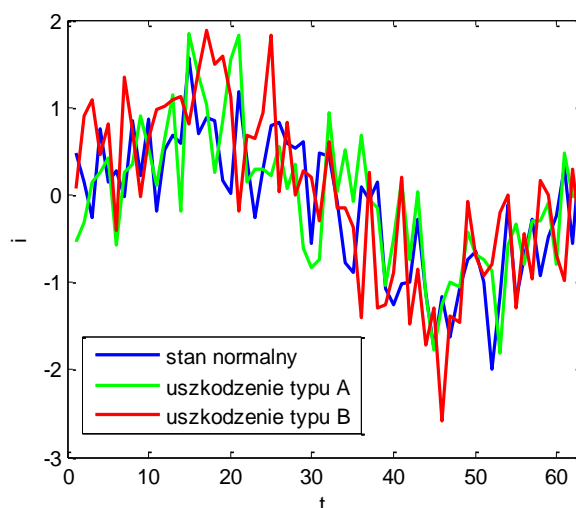
### Zadania pomocnicze

Zapoznaj się z funkcjami `knnsearch`, `crossval`, `predict` (help Matlab).

### Zadania do wykonania

Zadanie polega na zdiagnozowaniu urządzenia elektronicznego na podstawie przebiegu natężenia prądu chwilowego przepływającego przez to urządzenie (jeden okres sinusoidy). Na poniższym rysunku pokazano przykładowe przebiegi natężenia prądu dla trzech przypadków:

- normalny stan pracy urządzenia - linia niebieska,
- uszkodzenie typ A - linia zielona,
- uszkodzenie typu B - linia czerwona.



Zauważ, że przebiegi natężenia prądu są silnie zakłócone (przebieg idealny to sinusoida).

W bazie danych zgromadzono po 50 przebiegów natężenia prądu dla każdego przypadku (stan normalny, uszkodzenie A, uszkodzenie B). Przebiegi reprezentujące stan normalny oznaczmy etykietą klasy 1. Przebiegi reprezentujące uszkodzenie A oznaczmy etykietą klasy 2, a przebiegi reprezentujące uszkodzenie B oznaczmy etykietą klasy 3.

Zaprojektuj klasyfikator typu k najbliższych sąsiadów (k-NN) do rozpoznawania stanów urządzenia na podstawie przebiegów natężenia prądu. Parametr k dobierz w procedurze 10-krotnej krosvalidacji (wykład Uczenie się indukcyjne, slajd 15).

1. Wczytaj i zmodyfikuj zbiór danych:

```
load przebiegi_pradu; %załadowanie danych
rand('state',nr_gr*r_k);
x_trn = x + rand(150,63)*0.1-0.05; %dane treningowe
x_tst = x + rand(150,63)*0.1-0.05; %dane testowe
```

gdzie za `nr_gr` wstaw numer swojej sekcji a za `r_k` aktualny rok kalendarzowy.

Podejrzyj zmienne. Narysuj przebiegi prądu  $x_{trn}$  dla każdej klasy osobno. Numery klas zawarto w zmiennej  $c$ .

2. W procedurze 10-krotnej krosvalidacji znajdź optymalną wartość liczby najbliższych sąsiadów  $k$ :

```
for k=1:50 %liczbę sąsiadów zmieniamy od 1 do 50
    Mdl = fitcknn(x_trn,c,'NumNeighbors',k);
    CVMdl = crossval(Mdl,'Kfold',10);
    Acc(k) = 1 - kfoldLoss(CVMdl);
end
```

Zmienna  $Acc$  przechowuje procentową dokładność klasyfikatora oszacowaną w procedurze krosvalidacji. Narysuj wykres zależności  $Acc = f(k)$ . Wybierz optymalną wartość  $k$  i przypisz ją pod zmienną  $k_{best}$ .

3. Oblicz dokładność klasyfikatora dla danych testowych:

```
Mdl = fitcknn(x_trn,c,'NumNeighbors',k_best);
cp = predict(Mdl,x_tst);
Acc_tst = sum(c==cp)/150;
```

4. Powtórz obliczenia zmieniając miarę odległości z euklidesowej (domyślna) na miejską i Czebyszewa (patrz help dla funkcji `fitcknn`). Porównaj wyniki klasyfikacji dla trzech miar odległości.

## Co powinno znaleźć się w sprawozdaniu

- A) Cel ćwiczenia.
- B) Treść zadania.
- C) Opis używanej w ćwiczeniu metody  $k$  najbliższych sąsiadów (nie kopiuj treści wykładu, poszukaj w literaturze i Internecie).
- D) Metodyka rozwiązania – poszczególne instrukcje Matlaba z wynikami i komentarzem (zachowaj numerację zadań).
- E) Wnioski końcowe.

## Zadania dodatkowe dla ambitnych

1. W p. 2 zamiast 10-krotnej krosvalidacji użyj metody minus jednego elementu (*leave-one-out*). Porównaj tę metodę z 10-krotną krosvalidacją.
2. Wykonaj to ćwiczenie w innym środowisku, np. C/C++/C#, Python, R, ...

## Przykładowe zagadnienia i pytania zaliczeniowe

1. Cel i plan ćwiczenia.
2. Materiał ze sprawozdania.
3. Klasyfikator  $k$  najbliższych sąsiadów.
4. Metody selekcji atrybutów.
5. Metoda analizy głównych składowych.
6. Krosvalidacja.

## Do przygotowania na następne zajęcia

1. Zapoznać się z instrukcją do kolejnego ćwiczenia.

2. Zapoznać się z częścią teoretyczną do kolejnego ćwiczenia.
3. Wykonać zadania pomocnicze do kolejnego ćwiczenia.