Symbol ćwiczenia: KM

Klasyfikator minimalnoodległościowy

Część teoretyczna

Wykład 10 z SUS: Przekształcanie atrybutów.
Wykład 11 z SUS: Klasyfikatory minimalnoodległościowe.

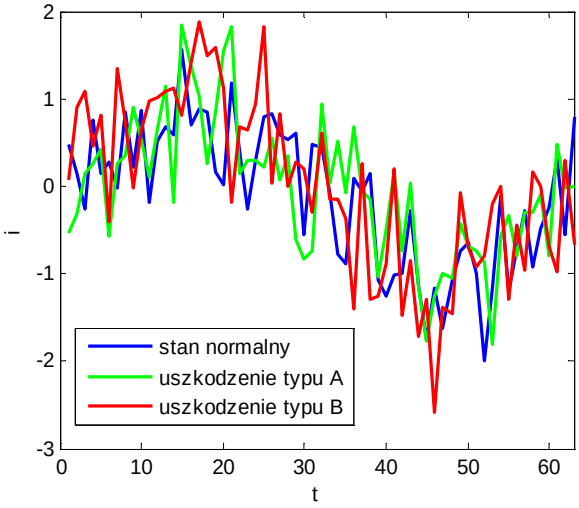
Zadania pomocnicze

Zapoznaj się z funkcjami `knnclassify`, `crossvalind`, `pca` (help Matlaba).

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 kroswalidacji (wykład 2 z SUS, slajd 15).

- Dokonaj selekcji atrybutów metodą krokowego dodawania atrybutów.

- Dokonaj ekstrakcji atrybutów metodą analizy głównych składowych (PCA). Sprawdź jaka jest dokładność klasyfikacji dla różnej liczby głównych składowych jako nowych atrybutów.

1. Wczytaj i zmodyfikuj zbiór danych:

```
load przebiegi_pradu; %załadowanie danych
```

```
rand('state',nr_gr*r_k);
x = x + rand(150,63)*0.1-0.05; %modyfikacja danych
```

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

Podejrzyj zmienne. Narysuj przebiegi prądu `x` dla każdej klasy osobno. Numery klas zawarto w zmiennej `c`.

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

```
id = crossvalind('Kfold',150,10); %podział przykładów na podzbiory
                                    do kroswalidacji
for k=1:20 %liczbę sąsiadów zmieniamy od 1 do 20
    for i = 1:10 %10-krotna kroswalidacja
        val = (id == i);
        train = ~val;
        c1=knnclassify(x(val,:),x(train,:),c(train),k,'euclidean');
        acc(i)=sum(c(val)==c1)/sum(val)*100; %dokładność
                                               klasyfikatora
    end
    Acc(k) = mean(acc); %średnie dokładności klasyfikatora przy
                         różnej liczbie najbliższych sąsiadów
end
```

Zmienna `Acc` przechowuje procentową dokładność klasyfikatora oszacowaną w procedurze kroswalidacji. Narysuj wykres zależności `Acc = f(k)`. Wybierz optymalną wartość `k` i przypisz ją pod zmienną `k_best`.

3. Przeprowadź selekcję atrybutów metodą sekwencyjnego dodawania atrybutów w procedurze 10-krotnej kroswalidacji według algorytmu (adaptacja algorytmu z wykładu 6 SUS, slajd 9):

1.
```
id = crossvalind('Kfold',150,10); %podział przykładów na podzbiory
                                    do kroswalidacji
acc_best = 0;
F = 1:63; %zbiór atrybutów kandydujących
O = []; %zbiór atrybutów istotnych
```
2. Powtarzaj dla każdego (*j*-tego) elementu zbioru `F`
   2.1. Utwórz tymczasowy zbiór `B` ze zbioru `O` i *j*-tego elementu zbioru `F`
   2.2. Zastosuj klasyfikator działający na atrybutach zawartych w zbiorze `B`:
```
for i = 1:10 %10-krotna kroswalidacja
      val = (id == i);
      train = ~val;
      c1=knnclassify(x(val,B),x(train,B),c(train),k_best,
      'euclidean');
      acc1(i)=sum(c(val)==c1)/sum(val)*100;
end
Acc1 = mean(acc1); %dokładność klasyfikatora przy bieżącym
                    zestawie atrybutów
```
   2.3. Jeśli `Acc1 > acc_best` przyjmij `acc_best = Acc1` i zapamiętaj *j*-ty atrybut: `j_best = j`
3. Jeśli w pętli 2 nie było poprawy rezultatu (`acc_best`) zakończ.
4. Dodaj `j_best` do zbioru `O` i usuń go ze zbioru `F`
5. Powtórz kroki 2-5

Jaki jest najlepszy zestaw atrybutów znaleziony przez ten algorytm (zbiór `O`)? Jaka jest dokładność klasyfikatora przy tym zestawie atrybutów? Czy nastąpiła poprawa w stosunku do klasyfikacji z pełnym zbiorem atrybutów?

4. Używając metody PCA (analizy głównych składowych) utwórz nowe atrybuty na podstawie oryginalnych. Zbadaj dokładność klasyfikatora, gdy do klasyfikacji użyjemy tylko pierwszej głównej składowej, dwóch pierwszych głównych składowych, ..., wszystkich głównych składowych.

   1. Ekstrakcja głównych składowych `xg`:

      ```
      [~,xg,variances] = princomp(x);
      ```

   2. Powtarzaj dla liczby głównych składowych `lg` równej od 1 do 63:
      2.1. Pobierz `lg` głównych składowych z tablicy `xg: xx=xg(:,1:lg)`
      2.2. Pętla 10-krotnej kroswalidacji j.w. dla `xx`
      2.3. Wyznaczenie dokładności średniej dla kolejnych wartości `lg`:
      `Acc2(lg) = mean(acc2)`

   Narysuj wykres zależności `Acc2 = f(lg)`. Jaka jest optymalna wartość liczby głównych składowych?

   Sporządź wykres wariancji wyjaśnionej przez kolejne główne składowe:

   ```
   pareto(explained);
   explained = 100*variances/sum(variances);
   xlabel('Numer głównej składowej');
   ylabel('Wyjaśniona wariancja (%)');
   ```

   Zobrazuj dwie pierwsze główne składowe przykładów:

   ```
   gscatter( xg(:,1),xg(:,2),c,'bgr','xo.');
   xlabel('Pierwsza główna składowa');
   ylabel('Druga główna składowa');
   ```

   Zobrazuj powierzchnie decyzyjne klasyfikatora opartego na dwóch pierwszych głównych składowych:

   ```
   [a,b] = meshgrid(-6:12/150:6);
   xq=[a(:), b(:)];
   c2=knnclassify(xq,xg(:,[1,2]),c,k_best,'euclidean');
   figure;
   gscatter(xq(:,1),xq(:,2),c2,'bgr','...');
   xlabel('Pierwsza główna składowa');
   ylabel('Druga główna składowa');
   ```

Co powinno znaleźć się w sprawozdaniu

A) Cel ćwiczenia.
B) Treść zadania.
C) Opis używanych w ćwiczeniu metod: k najbliższych sąsiadów, krokowe dodawanie cech, PCA (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.

Uwaga: Sprawozdania ze sztucznej inteligencji przesyłamy w postaci elektronicznej (pdf) pod nazwą

SI_*Nazwisko1*+*Nazwisko2*_*SymbolĆwiczenia*_*SymbolRoku*.pdf

gdzie:
SymbolĆwiczenia znajduje się na początku instrukcji (symbol tego ćwiczenia - KM),
SymbolRoku zawiera rok, symbol kierunku i oznaczenie "S" dla studiów stacjonarnych lub "NS" dla studiów niestacjonarnych; symbole oddzielamy znakiem "+" (*rok*+*kierunek*+*S*|*NS*)

Przykład prawidłowej nazwy pliku ze sprawozdaniem: SI_Nowak+Kowalski_KM_4+I+S.pdf
W temacie emaila proszę skopiować nazwę pliku ze sprawozdaniem j.w.

Sprawozdania niespełniające powyższych wymogów nie będą przyjmowane.

Zadania dodatkowe dla ambitnych

1. W p. 3 przeprowadź selekcję atrybutów metodą sekwencyjnej eliminacji atrybutów w procedurze 10-krotnej kroswalidacji. Porównaj wyniki z tymi otrzymanymi metodą dodawania atrybutów.
2. W p. 2 zamiast 10-krotnej kroswalidacji użyj metody minus jednego elementu (*leave-one-out*). Porównaj tę metodę z 10-krotną kroswalidacją.
3. 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. Kroswalidacja.

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.