ROB Laboratorium 2 - sprawozdanie

Mateusz Wasiak

9 kwietnia 2020

1 Ładowanie i zmiana etykiet

W ramach tego punktu wywołana została funkcja *load_cardsuits_data()*, która wczytała i przekształciła etykiety z 4 do 8.

2 Wartości odstające

W celu usunięcia wartości odstających korzystałem z funkcji liczących wartości maksymalne, minimalne oraz średnią i medianę. Wykorzystałem również funkcję *plot2features*.

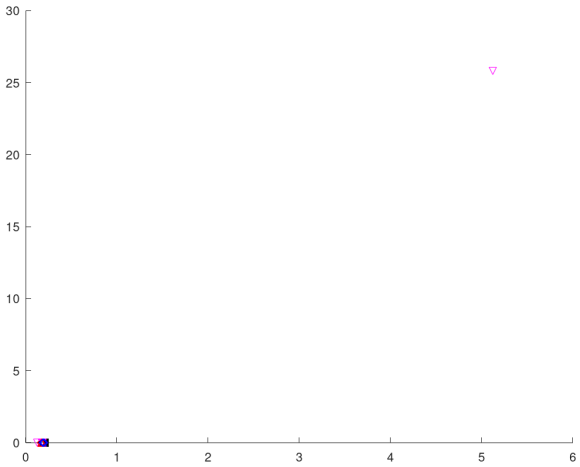
Poniższy kod oraz analiza wykresów (1, 2, 3) pozwoliły mi na usunięcie wartości odsdtających. (Próbki o indeksie 186 oraz próbki o indeksie 641).

```
[mean(train); median(train)]
[mv midx] = max(train)
midx = 186

train(midx-1:midx+1, :)
train(midx, :) = [];

[mv midx] = min(train)
midx = 641

train(midx-1:midx+1, :)
train(midx, :) = [];
```

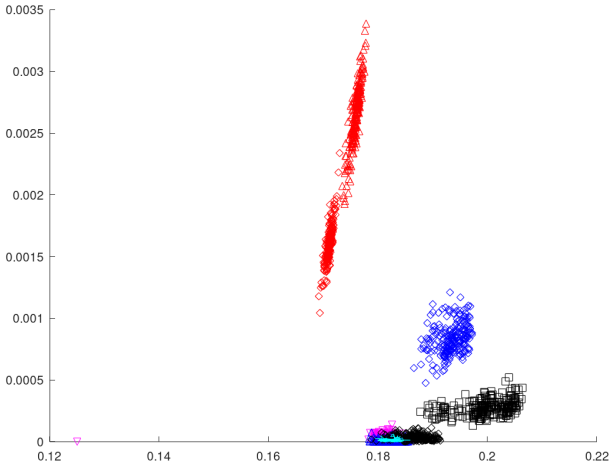
3 Klasyfikator Bayesa

3.1 Wybór cech

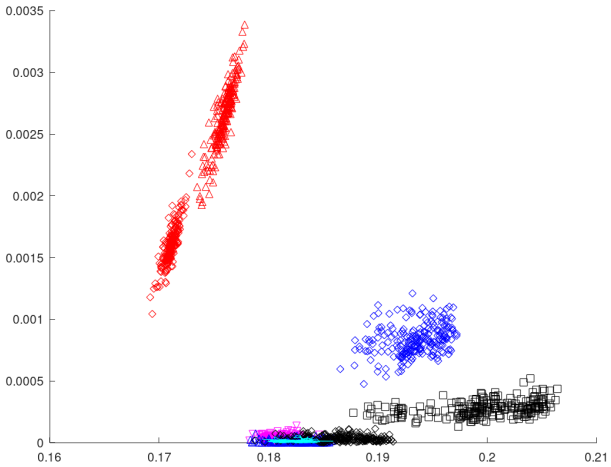
Po przejrzeniu większej liczby wykresów tworzonych przez *plot2features()* podjąłem decyzję o wyborze cech 2 i 3, ze względu na to, że klasy zostały dość dobrze odseparowane. Widać to na rysunku 4



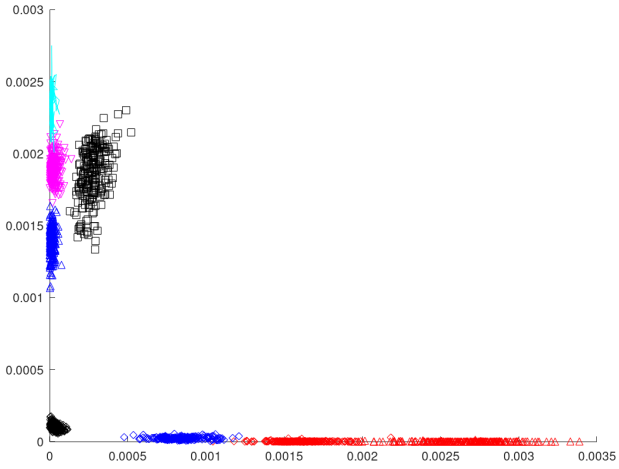
Rysunek 1: plot2features(train, 2, 3)



Rysunek 2: plot2features(train, 2, 3) po usunięciu punktu 186.



Rysunek 3: plot2features(train, 2, 3) po usunięciu punktu 641.



Rysunek 4: plot2features(train, 3,4) po usunięciu odstających danych.

3.2 Wyniki eksperymentu

Szerokośc okna dla klasyfikacji z oknem Parzena to 0.001

```
              %indep       %multi       %parzen
base_ercf = 0.021382     0.020833     0.016996
```

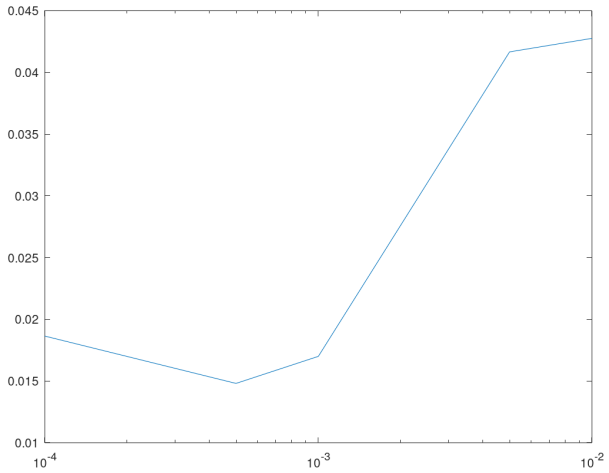
Klasyfikator z oknem Parzena pozwolił na osiągnięcie najlepszych wyników. Można zatem zaobserwować, że metoda która estymuje rozkład prawdopodobieństwa może najefektywniej dopasować się do rozkładu rzeczywistego. Na podstawie otrzymanych wyników można zauważyć, że rozkład wielowymiarowy daje odrobine lepsze wyniki niż rozkład uzyskany z kilku jednowymiarowych rozkładów

4 Reduckja zbioru uczącego

Test	Min.	Max.	Średnia	Odch. standardowe
0.1				
INDEP	0.0197368	0.0301535	0.0248904	0.0038102
MULTI	0.0186404	0.0279605	0.0243421	0.0037706
PARZEN	0.0356360	0.0389254	0.0367325	0.0013978
0.25				
INDEP	0.0180921	0.0235746	0.0205044	0.0020732
MULTI	0.0180921	0.0224781	0.0197368	0.0018183
PARZEN	0.0257675	0.0290570	0.0273026	0.0013089
0.5				
INDEP	0.0191886	0.0230263	0.0210526	0.0013761
MULTI	0.0180921	0.0213816	0.0203947	0.0013651
PARZEN	0.0191886	0.0230263	0.0206140	0.0016264

Tabela 1: Wyniki eksperymentu punkt 4

Na podstawie tabeli 1 można zaobserwować, że ilość próbek znajdujących się w zbiorze uczącym silnie wpływa na efektywność klasyfikacji. Najbardziej widoczne był to dla klasyfikacji z wykorzystaniem metody Parzena - klasyfikacja była znacznie bardziej nieefektywna dla małych zbiorów uczących. Pozostałe metody również osiągnęły słabsze wyniki, ale różnica nie jest aż tak duża.



Rysunek 5: Wykres zależności szekości okna h1 od stopy błędu.

5 Szerokość okna *h1*

h1	ercf
0.0001	1.8640e-02
0.0005	1.4803e-02
0.001	1.6996e-02
0.005	4.1667e-02
0.01	4.2763e-02

Tabela 2: Wyniki eksperymentu punkt 5

Na postawie tabeli 2 i wykresu 5 można wywnioskować, że jakoś klasyfikacji spada, gdy wybrane zostanie zbym duże lub zby małe okno.

6 Zmiana prawdopodobieństwa *apriori*

```
apriori = [0.165 0.085 0.085 0.165 0.165 0.085 0.085 0.165]
parts = [1.0 0.5 0.5 1.0 1.0 0.5 0.5 1.0];
```

Test	ercf
indep	0.01696
multi	0.0155
parzen	0.01301

Tabela 3: Wyniki eksperymentu punkt 6

```
ad6_cfmxs =
{
        [1,1] =

     228     0     0     0     0     0     0     0
       0   113     0     0     0     1     0     0
      10     0   101     0     0     0     3     0
       1     0     2   225     0     0     0     0
       0     0     0     0   227     0     0     1
       0     3     0     0     0   111     0     0
       1     0     2     0     0     0   111     0
       0     1     0     0     0     0     0   227

  [2,1] =

     228     0     0     0     0     0     0     0
       0   113     0     0     0     1     0     0
       8     0   103     0     0     0     3     0
       2     0     1   225     0     0     0     0
       0     0     0     0   227     0     0     1
       0     2     0     0     0   112     0     0
       0     0     3     0     0     0   111     0
       0     1     0     0     0     0     0   227

  [3,1] =

     224     0     1     3     0     0     0     0
       0   112     0     0     0     1     0     1
       2     0   106     1     0     0     5     0
       0     0     0   228     0     0     0     0
       0     0     0     0   228     0     0     0
       0     1     0     0     0   113     0     0
       0     0     3     0     0     0   111     0
       0     1     0     0     0     0     0   227
...
}
```

Na podstawie macierzy pomyłek klasyfikatora dla zadanego prawdopodobieństwa apriori i pełnego zioru z równym prawdopodobieństwem apriori można

zauważyć, że zmniejszenie liczby klasyfikowanych próbek w klasach w których zanotowano najwięcej błędów skutkuje zmniejszeniem stopy błędów.

7 Porównanie klasyfikatorów Bayesa oraz 1-NN

Normalizacja nie była wymagana, ze względu na zbliżone wartośći odchylenia standardowego i wartości średnich dla wybanych cech. Widoczne jest to również po przyjrzeniu się Wykresowi 4

```
ad7_ercf_1nn =   0.018092
ad7_confmx_1nn =
   228     0     0     0     0     0     0     0
     0   227     0     0     0     1     0     0
     1     0   211     2     0     2    12     0
     1     0     0   227     0     0     0     0
     0     0     0     0   228     0     0     0
     0     4     0     0     0   224     0     0
     0     0     8     0     0     0   220     0
     0     2     0     0     0     0     0   226
```

Można zauważyć, że klasyfikator 1-NN osiągnął lepsze wyniki od klasyfikatora Bayesa gdy gęstość była wyznaczana metodami *INDEP* oraz *MULTI*, natomiast osiągnął słabsze wyniki od klasyfikatora metodą okna Parzena.