

# Rozpoznawanie obrazów

## Laboratorium nr 3, 4

Tomasz Gałęcki

04 kwietnia 2019

### 1 Opis metody klasyfikacji

W tym zadaniu została użyta metoda *one vs. one*. Zestaw 45 klasyfikatorów liniowych w sposób jednogłośny klasyfikował skany cyfr (wszystkie 9 klasyfikatorów znających daną cyfrę musiało jednocześnie ją wskazać, aby głosowanie zostało uwzględnione; w p.p. próbka została oznaczana jako *odrzucona*).

### 2 Opis algorytmu wyznaczania parametrów płaszczyzny decyzyjnej

Do wyznaczenia parametrów płaszczyzny decyzyjnej została użyta metoda perceptronowa. Stała uczenia była wyliczana jako  $\eta(k) = \frac{1}{\sqrt{k}}$ . Uzyskane rezultaty wykorzystania wyuczonego zestawu klasyfikatorów:

<pre>&gt;&gt; % train set ans =     % good    % bad    % reject     0.922850    0.053000    0.024150</pre>	<pre>&gt;&gt; % test set ans =     % good    % bad    % reject     0.923000    0.053400    0.023600</pre>
--	---

### 3 Dane dotyczące jakości klasyfikacji każdego z wykorzystywanych klasyfikatorów liniowych

Każdy z klasyfikatorów został oceniony pod kątem trafności głosowań nad zbiorem **testowym**.

<pre>classifiersError([1 3], :) ans = {     [1,1] = % components         1    2         % err     [2,1] = 0.0014184     [1,2] =         1    3     [2,2] = 0.012922     [1,3] =         1    4     [2,3] = 0.0045226     [1,4] =         1    5     [2,4] = 0.0030581     [1,5] =         1    6     [2,5] = 0.011752     [1,6] =         1    7     [2,6] = 0.012384     [1,7] =</pre>	<pre>         1    8     [2,7] = 0.0049801     [1,8] =         1    9     [2,8] = 0.010747     [1,9] =         1   10     [2,9] = 0.0085470     [1,10] =         2    3     [2,10] = 0.0087679     [1,11] =         2    4     [2,11] = 0.010256     [1,12] =         2    5     [2,12] = 0.0023618     [1,13] =         2    6     [2,13] = 0.0064134     [1,14] =         2    7     [2,14] = 0.0052556</pre>	<pre>     [1,15] =         2    8     [2,15] = 0.0097087     [1,16] =         2    9     [2,16] = 0.013751     [1,17] =         2   10     [2,17] = 0.0055970     [1,18] =         3    4     [2,18] = 0.024486     [1,19] =         3    5     [2,19] = 0.017378     [1,20] =         3    6     [2,20] = 0.020790     [1,21] =         3    7     [2,21] = 0.019095     [1,22] =         3    8</pre>
---	---	---

[2,22] = 0.020874	[2,30] = 0.023279	[2,38] = 0.045016
[1,23] =	[1,31] =	[1,39] =
3 9	5 6	6 10
[2,23] = 0.023928	[2,31] = 0.012807	[2,39] = 0.015255
[1,24] =	[1,32] =	[1,40] =
3 10	5 7	7 8
[2,24] = 0.016659	[2,32] = 0.013402	[2,40] = 0.0045317
[1,25] =	[1,33] =	[1,41] =
4 5	5 8	7 9
[2,25] = 0.0070281	[2,33] = 0.013433	[2,41] = 0.012940
[1,26] =	[1,34] =	[1,42] =
4 6	5 9	7 10
[2,26] = 0.039432	[2,34] = 0.0076687	[2,42] = 0.0091510
[1,27] =	[1,35] =	[1,43] =
4 7	5 10	8 9
[2,27] = 0.0055894	[2,35] = 0.034154	[2,43] = 0.015485
[1,28] =	[1,36] =	[1,44] =
4 8	6 7	8 10
[2,28] = 0.022080	[2,36] = 0.026486	[2,44] = 0.046637
[1,29] =	[1,37] =	[1,45] =
4 9	6 8	9 10
[2,29] = 0.035282	[2,37] = 0.011458	[2,45] = 0.021684
[1,30] =	[1,38] =	}
4 10	6 9	

Najgorzej wypadł klasyfikator dla pary cyfr (7,9).

```
>> classifiersErrorTest(:, posWorstTest)
ans =
{
    [1,1] = % components
        8 10
    [2,1] = % confMx
        977 51
        44 965
    [3,1] = 0.046637 % error
}
```

## 4 Dane dotyczące jakości klasyfikacji cyfr, z wnioskami wynikającymi z analizy macierzy pomyłek

Analizując macierz pomyłek na zbiorze treningowym widać, że najwięcej błędnych głosów nastąpiło podczas rozróżniania cyfr 7 i 9. Na zbiorze testowym takich pomyłek też było dużo. Na zbiorze testowym najczęściej myloną parą były cyfry 5 i 9.

```
>> % train set
>> clab = unamvoting(tvec, ovo);
>> cfm = confMx(tlab, clab)
cfm =
    5697      0      22      11      10      52      30      5      18      7      71
      0    6490      31      28      8      10      3      23      67      6      76
      19      22    5457      47      50      19      47      41      69      8     179
      6      26      79    5495      0     179      9      40      73      31     193
      6      15      31      2    5465      3      21      20      14     147     118
      29      10      20     142     16    4786     54      13     107      22     222
      27      9      48      3      26      75    5616      0      22      0      92
      6      13      46      28      29      11      2    5863      8     127     132
      12      66      52      97      10     116      34      16    5188     43     217
      12      21      20      44     146      22      1     179      41    5314     149
```

```
>> % test set
>> clab = unamvoting(tstv, ovo);
>> cfm = confMx(tstl, clab)
    cfm =
    951     0     3     2     0     4     3     2     1     0    14
         0   1099     3     4     0     1     0     2    11     0    15
         4     1   952     7     6     1     7     7    17     2    28
         2     1    10   923     0    18     0    10    13     5    28
         2     0     5     1   920     0     6     4     4    23    17
         7     3     4    32     4   784     5     3    16     4    30
         9     3     9     1     6    16   901     0     1     0    12
         0     5    21     7     2     0     0   946     4    20    23
         4     3     3    22     5    27     7     4   856     3    40
         4     4     1     9    30     4     1    25     4   898    29
```

Po analizie poprawności klasyfikatorów niezależnie, to klasyfikator dla cyfr 7 i 9 okazał się najmniej trafny w zbiorze treningowym i testowym:

<pre>&gt;&gt; classifiersErrorTrain(:, posWorstTrain)     ans =     {         [1,1] = % components             8    10         [2,1] = % confMx             5991    274             267    5682         [3,1] = 0.044293 % error     }</pre>	<pre>&gt;&gt; classifiersErrorTest(:, posWorstTest)     ans =     {         [1,1] = % components             8    10         [2,1] = % confMx             977     51             44    965         [3,1] = 0.046637 % error     }</pre>
--	---