

ROB Laboratorium 34 - sprawozdanie

Mateusz Wasiak

6 maja 2020

1 Implementacja algorytmu wyznaczającego parametry płaszczyzny

Podczas tego zadania laboratoryjnego zaimplementowałem uczenie perceptronu [*perceptron.m*]. Wybrałem współczynnik uczenia jakos $1/\sqrt{i}$, ze względu na to, że pozwala on na znacznie dokładniejsze dopasowanie klasyfikatora niż pozostałe testowane współczynniki (np. $1/i$) w okolicach optimum.

2 Redukcja wymiarowości próbek

Redukcja wymiarowości próbek została wykonana przy pomocy funkcji *pcaTransform*. Poniższy kod przedstawia zmniejszenie wymiarowości próbek do 40 przy pomocy PCA (*ang. Principal Component Analysis*).

```
[mu trmx] = prepTransform(tvec , comp_count);  
tvec = pcaTransform(tvec , mu, trmx);  
tstv = pcaTransform(tstv , mu, trmx);
```

3 Klasyfikacja na podstawowym zbiorze danych

3.1 Zespół klasyfikatorów OVO (*one vs one*)

Uczenie zbioru klasyfikatorów One vs One zostało wykonane z wykorzystaniem kodu dołączonego w plikach do laboratorium. Wykonywane jest głosowanie jednomyślne - wszystkie 9 klasyfikatorów, które znają daną cyfrę musi zgadzać się co do klasy.

3.1.1 Wyniki eksperymentu

Dane	Poprawnie	Błędnie	Odrzucone
train	0.909383	0.063850	0.026767
test	0.912500	0.060400	0.027100

Tabela 1: Wyniki eksperymentu dla zbioru klasyfikatorów OVO

Macierze pomyłek:

$train_cfmx =$	5651	0	25	12	11	74	33	5	18	5	89
	0	6453	43	25	9	13	1	26	85	7	80
	27	19	5366	53	56	24	80	46	83	17	187
	9	23	79	5436	0	195	15	36	104	41	193
	9	16	42	1	5386	4	28	24	20	183	129
	34	16	28	178	26	4664	51	10	129	37	248
	29	9	88	4	29	83	5554	0	24	0	98
	8	20	57	31	38	10	1	5797	13	152	138
	13	81	64	129	19	128	28	17	5057	48	267
	11	19	37	60	171	22	1	206	46	5199	177
	945	0	4	1	0	8	6	1	2	0	13
	0	1093	3	3	0	2	0	1	15	0	18
$test_cfmx =$	5	2	932	8	6	3	11	10	17	3	35
	1	0	12	916	0	25	1	11	12	6	26
	1	0	5	1	912	0	7	5	4	29	18
	7	3	3	36	5	771	4	1	24	5	33
	9	2	10	0	8	11	893	1	3	0	21
	0	6	20	5	4	0	0	935	6	24	28
	4	1	4	23	5	26	6	4	855	5	41
	6	5	4	8	34	4	1	27	9	873	38

Poniżej znajdują się wyniki poprawnych klasyfikacji dla poszczególnych par klasyfikowanych liczb:

Para liczb	współ.	Para liczb	współ.	Para liczb	współ.
1 i 2	0.99858	2 i 9	0.98103	5 i 6	0.98879
1 i 3	0.98509	2 i 0	0.99487	5 i 7	0.98454
1 i 4	0.99397	3 i 4	0.97307	5 i 8	0.98408
1 i 5	0.99694	3 i 5	0.98163	5 i 9	0.98722
1 i 6	0.98237	3 i 6	0.97245	5 i 0	0.95881
1 i 7	0.98349	3 i 7	0.97337	6 i 7	0.97405
1 i 8	0.99353	3 i 8	0.97524	6 i 8	0.99115
1 i 9	0.98925	3 i 9	0.96810	6 i 9	0.95338
1 i 0	0.99296	3 i 0	0.98285	6 i 0	0.97948
2 i 3	0.98616	4 i 5	0.99398	7 i 8	0.98993
2 i 4	0.99114	4 i 6	0.95321	7 i 9	0.98551
2 i 5	0.99764	4 i 7	0.99289	7 i 0	0.98882
2 i 6	0.99013	4 i 8	0.98086	8 i 9	0.98202
2 i 7	0.99474	4 i 9	0.96371	8 i 0	0.94944
2 i 8	0.99029	4 i 0	0.97623	9 i 0	0.96924

3.1.2 Wnioski

Można zaobserwować, że efekt przeuczenia nie wystąpił, wyniki na zbiorze testowym są nawet odrobine lepsze niż na zbiorze treningowym. Na podstawie analizy macierzy pomyłek można zauważyć, że najwięcej błędów występuje dla pary 8 i 0. Najprawdopodobniej jest to spowodowane przez dość duże wizualne podobieństwo pomiędzy tymi dwoma cyframi. Pary, które znacznie różnią się wyglądem osiągają znacznie lepsze wyniki. Analogiczne wnioski można wysnuć przyglądając się jakości poszczególnych klasyfikatorów.

3.2 Zespół klasyfikatorów OVR (*one vs rest*)

W ramach tej części laboratorium zaimplementowałem dwie funkcje - *trainOVRensemble* i *unamvoting_ovr*. Pierwsza z nich jest bardzo podobna do *trainOVOensemble* - z tą różnicą, że szkolenie będzie odbywać się nie dla każdej pary, a dla wartości lub reszta. Funkcja *unamvoting_ovr* wykorzystuje tę samą metodę głosowania co *unamvoting* (głosowanie jednomyślne), ale jest dostosowana do *one vs rest*.

3.2.1 Wyniki eksperymentu

Dane	Poprawnie	Błędnie	Odrzucone
train	0.767400	0.037650	0.194950
test	0.773300	0.035000	0.191700

Tabela 2: Wyniki eksperymentu dla zbioru klasyfikatorów OVR

Macierze pomyłek:

$train_cfmx =$	5232	0	5	4	2	19	16	2	3	1	639
	0	6014	23	10	3	19	1	5	55	5	607
	8	19	4529	35	36	12	32	27	72	9	1179
	18	8	62	4568	1	129	14	17	52	38	1224
	6	8	11	3	4460	10	11	9	41	78	1205
	27	17	17	111	50	3336	44	6	53	37	1723
	12	13	22	1	22	47	5020	1	19	2	759
	13	8	50	7	24	16	1	5074	10	79	983
	23	64	18	70	3	53	20	7	3830	24	1739
	15	11	12	59	87	25	0	89	31	3981	1639
$test_cfmx =$	861	0	0	1	0	3	1	1	0	0	113
	0	1027	1	1	0	0	3	1	9	0	93
	1	2	761	7	5	1	4	6	18	1	226
	2	0	4	784	0	17	2	4	9	6	182
	2	0	0	0	770	1	1	1	8	16	183
	5	2	1	28	5	563	7	2	12	2	265
	8	2	3	1	2	8	819	0	4	0	111
	2	1	19	0	2	0	0	819	3	5	177
	7	4	3	11	5	5	5	2	652	2	278
	2	4	2	2	18	5	0	9	1	677	289

Poniżej znajdują się wyniki poprawnych klasyfikacji dla poszczególnych klas:

Klasa poddana klasyfikacji	współczynnik
1	0.98940
2	0.99200
3	0.97390
4	0.97020
5	0.97440
6	0.95520
7	0.98390
8	0.98230
9	0.95180
0	0.95550

Tabela 3: Wyniki poprawnych klasyfikacji dla poszczególnych klas klasyfikatorów OVR

3.2.2 Wnioski

Klasyfikacja z wykorzystaniem zespołu klasyfikatorów one vs rest ma znacznie niższą jakość niż wykorzystany we wcześniejszym punkcie zespół klasyfikatorów one vs one.

Dwukrotnie zmalała liczba błędów, ale znacznie zrosła liczba odrzuconych prób klasyfikacji. Najprawdopodobniej jest to spowodowane przez jednomyślne głosowanie, które wymaga by wszystkie poza jednym klasyfikatory wskazały liczbę jako nierozpoznaną, co w przypadku podobnych do siebie liczb może być bardzo trudne.

4 Roszerzenie zbioru i klasyfikacja

W celu rozszerzenia zbioru cech została wykorzystana została dostarczona w plikach laboratorium funkcja *expandFeatures*. Funkcja ta dodaje cechy będące iloczynami oryginalnych cech. Dla 40 początkowych atrybutów końcowo otrzymaliśmy ich aż 860. Na rozszerzonym zbiorze zostały wykonane testy z wcześniejszych punktów.

4.1 Wyniki eksperymentów

Dane	Poprawnie	Błędnie	Odrzucone
train	0.9681833	0.0244333	0.0073833
test	0.958500	0.030500	0.011000

Tabela 4: Wyniki eksperymentu dla rozszerzonego zbioru cech zespołu klasyfikatorów OVO

Macierz pomyłek dla zbioru testowego przy rozszerzonym zbiorze cech zespołu klasyfikatorów OVO:

$$test_{cfmx} = \begin{vmatrix} 962 & 0 & 1 & 2 & 0 & 0 & 5 & 1 & 3 & 1 & 5 \\ 0 & 1113 & 2 & 1 & 1 & 0 & 3 & 0 & 5 & 1 & 9 \\ 3 & 1 & 982 & 7 & 2 & 1 & 6 & 10 & 9 & 2 & 9 \\ 0 & 0 & 4 & 966 & 0 & 7 & 0 & 10 & 5 & 5 & 13 \\ 2 & 1 & 3 & 1 & 939 & 0 & 5 & 0 & 2 & 13 & 16 \\ 3 & 1 & 1 & 11 & 1 & 844 & 5 & 5 & 5 & 5 & 11 \\ 9 & 4 & 1 & 1 & 7 & 4 & 924 & 0 & 3 & 0 & 5 \\ 1 & 6 & 11 & 3 & 2 & 0 & 0 & 971 & 4 & 14 & 16 \\ 2 & 0 & 4 & 11 & 1 & 5 & 1 & 5 & 931 & 3 & 11 \\ 1 & 5 & 3 & 9 & 11 & 1 & 1 & 9 & 1 & 953 & 15 \end{vmatrix}$$

Dane	Poprawnie	Błędnie	Odrzucone
train	0.918083	0.016233	0.065683
test	0.918000	0.018800	0.063200

Tabela 5: Wyniki eksperymentu dla rozszerzonego zbioru cech zespołu klasyfikatorów OVR

Klasa poddana klasyfikacji	współczynnik
1	0.99600
2	0.99550
3	0.98880
4	0.98810
5	0.98930
6	0.98830
7	0.99100
8	0.98890
9	0.98800
0	0.98370

Tabela 6: Wyniki eksperymentu dla zbioru klasyfikatorów OVR przy rozszerzonym zbiorze cech

4.2 Wnioski

Rozszerzenie zbioru atrybutów przyczyniło się do znacznego wzrostu jakości klasyfikacji. Najbardziej widoczny jest zysk dla klasyfikatorów one vs rest. Taka poprawa wynika najprawdopodobniej z tego, że zwiększenie liczby cech i ich wymiarowości umożliwiło lepsze odseparowanie od siebie klas.

5 Usprawnienie rozwiązania standardowego

W kontekście usprawnienia rozwiązania rozważałem kilka podejść.

Jednym z nich mogłoby być zmiana głosowania jednomyślnego na wybór dwóch najlepiej ocenianych liczb a następnie wykonanie klasyfikacji pomiędzy dwiema tymi liczbami - to całkowicie zredukowałoby odrzucenie i prawdopodobnie zwiększyłoby liczbę poprawnych ale i liczbę błędnych klasyfikacji.

Drugim z nich mogłoby być stworzenie zespołu klasyfikatorów two vs rest. Na podstawie wyników tego zespołu klasyfikatorów wykonywane byłyby odpowiednie klasyfikacje one vs one dla tych przypadków, klasyfikatorów two vs rest sklasyfikowałby, że należy ona do zbioru two.

Trzecim podejściem była by modyfikacja drugiego podejścia. Wyniki nie odrzuconej klasyfikacji według zespołu klasyfikatorów one vs one byłyby jeszcze sprawdzane przez zespół klasyfikatorów two vs rest, gdzie wszystkie pary liczb byłyby klasyfikatorami, które jednogłośnie potwierdziły klasyfikację dla one vs one.

Niestety nie wykonałem implementacji powyższych rozwiązań.