Uczenie maszyn
PROJEKT ⋆

Przemysław Widz and Tomasz Zawadzki

Wrocław University of Science and Technology, 27 Wybrzeże Wyspiańskiego st., Wrocław

Streszczenie Zespoły heterogenicznych klasyfikatorów. Głosowanie większościowe i akumulacja wsparć (głosowanie miękkie i twarde).

Keywords: machine learning · classifiers · soft voting · hard voting · ensemble learning

1 Cel eksperymentu

Celem eksperymentu jest przeprowadzenie analizy porównawczej pomiędzy głosowaniem miękkim (akumulacja wsparć) i twardym (głosowanie większościowe) dla zespołów klasyfikatorów heterogeniczych w różnych konfiguracjach.

2 Plan eksperymentu

Podczas eksperymentu zostaną stworzone trzy różne zespoły klasyfikatorów. Klasyfikatory użyte do ich budowy to:

- *KNeighborsClassifier*
- *DecisionTreeClassifier*
- *LogisticRegression*
- *GaussianNB*
- *SupportVectorMachnie (SVC)*

Skład tworzonych zespołów przedstawiono poniżej:

1. **Zespół 1**
   - *KNeighborsClassifier*
   - *DecisionTreeClassifier*
   - *LogisticRegression*
2. **Zespół 2**
   - *LogisticRegression*
   - *GaussianNB*
   - *SVC*
3. **Zespół 3**
   - *KNeighborsClassifier*
   - *DecisionTreeClassifier*
   - *LogisticRegression*
   - *GaussianNB*
   - *SVC*

⋆ Sprawozdanie projektowe przygotowane z wykorzystaniem edytora LATEX w formacie LNCS. `https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines?countryChanged=true`

Parametry poszczególnych klasyfikatorów pozostaną niezmienione - wartości parametrów zostaną ustawione na domyślne, przedstawione w dokumentacji [1]

Wyjątkiem będą klasyfikatory LogisticRegression (parametr *max_iter* będzie wynosił 1000000) oraz SVC (parametr *gamma* będzie miał wartość **auto** oraz parametr *probability* ustawiony zostanie na wartość **True**). Tak stworzone zespoły zostaną przekazane do klasyfikatora VotingClassifier [1].

Eksperymenty zostaną przeprowadzone na 20 różnych zbiorach danych, opisanych w rozdziale 3. Odpowiednie odczytanie oraz przekształcenie danych zawartych w zbiorach będzie możliwe dzięki bibliotece *numpy* [3]. Zbiory danych będą dwukrotnie dzielone na pięć podzbiorów za pomocą stratyfikowanej walidacji krzyżowej.

Na podstawie dostarczonych przez te zbiory danych treningowych oraz testowych następi kolejno trenowanie oraz predykcja. Obie operacje zostaną przeprowadzone dla akumulacji wsparć (głosowanie miękkie) oraz głosowania większościowego (głosowanie twarde). W tabelach przedstawiających wyniki przeprowadzonych eksperymentów umieszczona zostanie średnia dokładność klasyfikacji uzyskana dla poszczególnych zespołów klasyfikatorów i określonych zbiorów danych.

Ostatnim krokiem będzie przeprowadzenie testów statystycznych. Dla każdego z 20 zbiorów danych przeprowadzone zostaną parowe testy statystyczne mające na celu porównanie między sobą zespolów klasyfikatorów każdy z każdym i ewentualne wskazanie w każdej z par, czy któryś z modeli jest statystycznie znacząco lepszy od drugiego. Do przeprowadzenia tego eksperymentu wykorzystany zostanie test *T Studenta* [4]. W celu wykonania testu na wielu zbiorach danych zostanie użyty parowy test *Wilcoxona* [4].

3 Zbiory danych

Zbiory danych wykorzystywane w eksperymentach zostały pobrane w większości z repozytorium *KEEL* [5]. Tylko jeden zbiór danych o nazwie **hepatitis** pochodzi ze strony *UCI Machine Learning repository* [6]. Nazwy zbiorów danych oraz liczba cech i klas w nich zawartych przedstawia poniższa tabela.

Tablica 1: Zbiory danych wykorzystywane w projekcie

Nazwa	**Liczba instancji**	**Liczba cech**	**Liczba klas**
hepatitis	155	19	2
winequality-red	1599	11	11
vowel	990	13	11
titanic	2201	3	2
spectfheart	267	44	2
movement_libras	360	90	15
segment	2310	19	7
led7digit	500	7	10
zoo	101	16	7
phoneme	5404	5	2
wdbc	569	30	2
yeast	1484	8	10
winequality-white	4898	11	11
twonorm	7400	20	2
ring	7400	20	2
texture	5500	40	11
vehicle	846	18	4
haberman	306	3	2
monk-2	432	6	2
balance	625	4	3

4 Wyniki przeprowadzonych eksperymentów

Tablica 2: Wyniki eksperymetrów oraz testy statystyczne dla głosowania twardego

Zbiór danych	**Zespół 1**	**Zespół 2**	**Zespół 3**
hepatitis	0.831 -	0.887 -	0.869 -
winequality-red	0.843 2,3	0.811 -	0.812 -
vowel	0.880 2	0.780 -	0.893 2
titanic	0.788 2,3	0.783 -	0.783 -
spectfheart	0.788 -	0.820 1,3	0.802 -
movement_libras	0.769 2	0.629 -	0.754 2
segment	0.972 2	0.929 -	0.969 2
led7digit	0.709 -	0.718 -	0.718 -
zoo	0.955 -	0.955 -	0.965 -
phoneme	0.877 2,3	0.795 -	0.851 2
wdbc	0.951 -	0.939 -	0.948 2
yeast	0.579 2	0.495 -	0.573 2
winequality-white	0.578 2	0.549 -	0.595 1,2
twonorm	0.974 -	0.978 1	0.978 1
ring	0.828 -	0.847 1	0.844 1
texture	0.985 2,3	0.964 -	0.977 2
vehicle	0.758 2,3	0.625 -	0.729 2
haberman	0.735 -	0.743 -	0.735 -
monk-2	0.992 2,3	0.915 -	0.972 2
balance	0.855 -	0.903 1	0.898 1
Average rank	2.15 -	1.75 -	2.1 -

Tablica 3: Wyniki eksperymentów oraz testy statystyczne dla głosowania miękkiego

Zbiór danych	**Zespół 1**	**Zespół 2**	**Zespół 3**
hepatitis	0.825 -	0.887 -	0.875 -
winequality-red	0.840 2	0.811 -	0.843 2
vowel	0.856 -	0.837 -	0.918 1,2
titanic	0.788 2,3	0.776 -	0.783 2
spectfheart	0.781 -	0.809 1	0.802 -
movement_libras	0.746 2	0.651 -	0.772 2
segment	0.977 2	0.933 -	0.973 2
led7digit	0.720 -	0.703 -	0.714 -
zoo	0.946 -	0.960 -	0.960 -
phoneme	0.886 2	0.797 -	0.871 2
wdbc	0.953 -	0.944 -	0.949 -
yeast	0.534 2	0.397 -	0.565 1,2
winequality-white	0.616 2	0.565 -	0.629 1,2
twonorm	0.968 -	0.979 1,3	0.976 1
ring	0.858 -	0.981 1,3	0.942 1
texture	0.981 2,3	0.942 -	0.975 2
vehicle	0.735 2	0.707 -	0.739 2
haberman	0.680 -	0.742 1	0.730 1
monk-2	0.993 2,3	0.972 -	0.972 -
balance	0.824 -	0.906 1,3	0.879 1
Average rank	1.95 -	1.7 -	2.35 2

5 Podsumowanie

Uzyskane średnie wyniki dokładności klasyfikacji dla poszczególnych zbiorów danych oraz zespołów klasyfikatorów i metod głosowania osiągają wyniki znacznie przekraczające próg 50%. Jedynie w przypadku zbiorów danych *yeast* oraz *winequality-wite*, zarówno w przypadku głosowania twardego jak i miękkiego dokładność klasyfikacji zbliża się lub nawet spada poniżej progu dokładności 50%.

Z analizy uśrednionych rang można odczytać, że dla przeprowadzonych badań na wybranych przez nas zbiorach danych najlepsze wyniki dla głosowania twardego uzyskał zespół 1, a dla głosowania miękkiego zespół 3.

Mimo różnic, jakie występują pomiędzy przetestowanymi przez nas zespołami klasyfikatorów, eksperymenty przeprowadzone na 20 zbiorach danych nie pozwalają stwierdzić, który zespół klasyfikatorów dla głosowania **twardego** jest lepszy z istotnością statystyczną. Natomiast w przypadku głosowania **miękkiego** zespół 3 (KNeighborsClassifier, DecisionTreeClassifier, LogisticRegression, GaussianNB, SVC) jest statystycznie znacząco lepszy od zespołu 2 (LogisticRegression, GaussianNB, SVC).

Literatura

1. *Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.*
2. *McKinney, W., others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).*
3. *Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2*
4. *Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272.*
5. *KEEL-dataset citation paper: J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17:2-3 (2011) 255-287.*
6. *Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.*