Optymalizacja wielkości bootstrapu, w zależności od klasyfikatora bazowego

T. Kołodziej
Wydział Informatyki i Telekomunikacji
Politechnika Wrocławska
Wrocław, Polska
251004@student.pwr.edu.pl

M. Kubiak
Wydział Informatyki i Telekomunikacji
Politechnika Wrocławska
Wrocław, Polska
249481@student.pwr.edu.pl

***Streszczenie*—15: Przedstawienie finalnej wersji projektu i omówienie rezultatów. [2h] [30.01.23r.]**

***Index Terms*—Bagging, Multi-Later Perceptron(MLP), Support Vector Machine(SVM), Gauss Naive Bayes(GNB), Drzewo decyzyjne**

I. Wstęp

Jednym z zadań postawionych w dziedzinie uczenia maszynowego jest klasyfikacja obiektów. Polega ono na przypisaniu etykiety do przykładu z dziedziny danego problemu, wykorzystując algorytmy uczenia maszynowego. Zadanie przydzielania etykiet zostaje wykonane przez klasyfikatory, które po wstępnym wyuczeniu na testowym zbiorze danych, są w stanie samodzielnie określić klasę danych wejściowych. W zależności od zasady działania, cechy klasyfikatorów mogą zostać uznane za zalety lub też wady, uwzględniając szybkość działania, dokładność czy skłonność do przeuczenia.

Multi-Layer Perceptron(MLP) jest w stanie zapewnić wyższą sprawność w porównaniu z innymi klasyfikatorami, jednak charakteryzują go również długie czasy trenowania oraz możliwość utknięcia w minimach lokalnych [1]. Funkcją decyzyjną Support Vector Machine(SVM) jest hiperpłaszczyzna oddzielająca obserwacje należące do różnych klas, na podstawie wzorców informacji o tych obserwacjach, zwanych cechami. Płaszczyzna optymalna to taka, która separuje klasy z możliwie największym marginesem [2]. Gauss Naive Bayes(GNB) to probabilistyczny algorytm kwalifikacji, cieszący się dużą popularnością, dzięki prostocie, sprawności i solidności działania [3]. Drzewa decyzyjne klasyfikują dane na podstawie serii pytań, ułożonej w hierarchii, gdzie pytania posiadają określoną liczbę odpowiedzi, z których każda przekierowuje decyzję do kolejnego, podrzędnego pytania, tworząc tym samym strukturę drzewa [4].

W celu zniwelowania niekorzystnych cech klasyfikatorów, zostają one złożone w zespoły, składające się z wielu klasyfikatorów o mniej skomplikowanej strukturze, niż byłoby to wymagane w przypadku pojedynczego klasyfikatora. Zespoły klasyfikatorów, takie jak Bagging lub Boosting, trenują klasyfikatory bazowe wykorzystując różne zestawy danych. W przypadku Baggingu, wynik zostaje wybrany na podstawie głosowania większościowego spośród wszystkich klasyfikatorów bazowych. Dzięki wytrenowaniu ich różnymi zestawami danych, zwiększona zostaje zdolność do generalizacji modelu, jednocześnie zwiększając jego odporność na przetrenowanie [5].

Celem naszej pracy jest optymalizacja wielkości bootstrapu w zależności od danego klasyfikatora bazowego, przy zastosowaniu heterogenicznego Baggingu. We wstępnym etapie zostanie przeprowadzona indywidualna optymalizacja bootstrapu dla klasyfikatorów bazowych MLP, SVM, GNB i drzewa decyzyjnego. Następnie przeprowadzona zostanie optymalizacja bootstrapu zespołu klasyfikatorów, zawierającego wcześniej wspomniane algorytmy klasyfikacji.

II. Powiązane prace

Artykuł [1] opisuje klasyfikator MLP oraz łagodzeniu jego wad przez operowanie w zespole klasyfikatorów. SVM zostaje dokładnie opisany z podziałem na etapy analizy danych w [2], opierając się na przykładzie neuroobrazowania mózgu. [3] jest pracą poświęconą predykcji rezygnacji klientów, za pomocą algorytmu GNB, wprowadzając wykorzystanie równoległych obliczeń wartości średniej, odchylenia standardowego i prawdopodobieństwa, w celu zwiększenia szybkości działania algorytmu. W artykule [6], drzewa decyzyjne i SVM zostały przedstawione jako odpowiednie dla większości zadań klasyfikacji, w formie krótkiego przeglądu danych. Drzewa decyzyjne zostały również opisane w pracy [4], omawiając zasadę ich działania, rodzaje problemów, do których mogą one zostać zastosowane oraz podając ich zastosowania w dziedzinie biotechnologii. W ramach publikacji [7] zaprezentowano powstawanie, rozwój oraz zasady działania drzew decyzyjnych oraz metod Boosting i Bagging. W artykule [5], Boosting przedstawiono jako prosty sposób na ograniczenie występowania problemu błędnej klasyfikacji i zmniejszenie wielkości zestawu uczącego, potrzebnego do osiągnięcia wyników porównywalnych do przypadku użycia pojedynczych klasyfikatorów. Artykuł [8] zagłębia się w temat heterogenicznych zespołów klasyfikatorów, badając wpływ różnorodności zestawów uczących i kompozycji bazowych klasyfikatorów w zespole, w celu osiągnięcia optymalnych wyników klasyfikacji.

III. Założenia projektowe

A. Research questions

- Jakie wpływ na jakość klasyfikacji ma dobór parametrów klasyfikatora?
- Jaki wpływ ma wielkość bootstrapu na jakość klasyfikacji bazowych klasyfikatorów?
- W jakim stopniu bagging klasyfikatorów bazowych pozwala na otrzymanie wyższej skuteczności klasyfikacji?

B. Cele eksperymentu

- Optymalizacja parametrów klasyfikatorów bazowych dla danego zadania klasyfikacji
- Optymalizacja parametrów klasyfikatorów bazowych dla generalnych zadań klasyfikacji
- Optymalizacja wielkości boostrapu w zależności od danego klasyfikatora bazowego
- Optymalizacja wielkości bootstrapu dla zespołu klasyfikatorów.
- Analiza otrzymanych wyników eksperymentów.

C. Opis zestawu danych

Badania zostaną przeprowadzone dla zestawu danych opisującego rodzaje grzybów.

Tabela I
Parametry zestawu danych

Zestaw danych grzybów	
Rodzaj zadania	*Klasyfikacja*
Pochodzenie	*Świat rzeczywisty*
Liczba cech	*22*
Rzeczywiste/Całkowite/Nominalne	*0/0/22*
Klasy	*2*
Liczba instancji	*5644*

Każda z 5644 instancji została opisana przez 22 cechy o wartościach nominalnych. Zadanie klasyfikacji polega na przypisaniu danej instancji jednej z dwóch klas, oznaczającej jadalność danego grzyba (poisonous/edible).

Tabela II
Cechy zestawu danych

Atrybut	**Dziedzina**
Cap-shape	*x, b, s, f, k, c*
Gill-color	*k, n, g, p, w, h, u, e, b, r, y, o*
Veil-type	*p, u*
Cap-surface	*s, y, f, g*
Stalk-shape	*e, t*
Veil-color	*w, n, o, y*
Cap-color	*n, y, w, g, e, p, b, u, c, r*
Stalk-root	*e, c, b, r, u, z*
Ring-number	*o, t, n*
Bruises	*t, f*
Stalk-surface-above-ring	*s, f, k, y*
Ring-type	*p, e, l, f, n, c, s, z*
Odor	*p, a, l, n, f, c, y, s, m*
Stalk-surface-below-ring	*s, f, y, k*
Spore-print-color	*k, n, u, h, w, r, o, y, b*
Gill-attachment	*f, a, d, n*
Stalk-color-above-ring	*w, g, p, n, b, e, o, c, y*
Population	*s, n, a, v, y, c*
Gill-spacing	*c, w, d*
Stalk-color-below-ring	*w, p, g, b, n, e, y, o, c*
Habitat	*u, g, m, d, p, w, l*
Gill-size	*n, b*
Class	*p, e*

IV. Plan eksperymentu

A. Opis środowiska eksperymentalnego

Środowisko badawcze zrealizowane zostanie przy pomocy IDE PyCharm Community Edition 2022.3, będącego otwartym oprogramowaniem, umożliwiającym tworzenie i kompilowanie skryptów.

Oprogramowanie testowe stworzone zostanie w języku Python w wersji 3. Tworzenie i szkolenie modeli ML oraz analiza ich skuteczności i wyników zostanie zrealizowana dzięki bibliotece scikit-learn, zbudowanej na bibliotekach *NumPy, SciPy i matplotlib*.

B. Dokładny opis planu eksperymentu (z naciskiem na wykorzystywany protokół badawczy)

W pierwszym etapie zostanie przeprowadzona optymalizacja parametrów klasyfikatorów bazowych - SVM, GNB, MLP, Drzewa decyzyjne. Celem eksperymentu jest przeprowadzenie przeszukiwania Grid Search oraz Monte Carlo dla wygnerowanego, syntetycznego zbioru danych oraz wykorzystaniu wyników jakości klasyfikacji do zdeniniowania parametrów, przy których osiągnięte zostaną najlepsze wyniki.

W następnym badaniu przeprowadzony zostanie ten sam eksperyment, na większej puli zbiorów. Celem badania jest zdefiniowanie parametrów klasyfikatorów do rozwiązywania generalnych problemów klasyfikacji. Na podstawie informacji o skuteczności klasyfikatorów na tle różnorodnych zbiorów danych, określone zostaną optymalne ustawienia.

W projekcie zbudowane zostaną cztery różne modele, wykorzystujące ten sam schemat działania, przyjmując na wejście bootstrap odpowiedniego rozmiaru i dokonując ewaluacji klasy. Dla modeli przeprowadzona zostanie walidacja krzyżowa, dzieląc zbiór na uczący i testowy. Początkowo modele zostaną przetrenowane na pełnym zbiorze danych. Po wytrenowaniu otrzymany zostanie współczynnik skuteczności klasyfikacji, po czym badanie zostanie powtórzone przy zmiejszającej się liczbie próbek. Wielkość bootstrapu będzie stopniowo zmniejszana, dzięki czemu uzyskana zostanie informacja na temat, wpływu zmian na dokładność klasyfikacji. Wartości te będą stopniowo zmniejszane, a na koniec porównane ze sobą i zostanie wybrana optymalna wielkość podzbioru, dla której osiągnięto najlepszy stosunek jakości klasyfikatora i rozmiaru bootstrapu.

W ostatnim etapie eksperymentu zostanie przeprowadzona integracja poszczególnych klasyfikatorów przez wykorzystanie baggingu. W takim klasyfikatorze złożonym, modele określają klasy obiektów osobno, po czym przeprowadzone zostaje głosowanie większościowe. Wynik głosowania spowoduje przypisanie pojedynczej klasy do instancji problemu.

V. Wyniki Badań i Analizy Statystycznej

12-14: Przeprowadzenie badań, opracowanie wyników oraz analiza statystyczna. [6h] [30.01.23r.]

...

VI. Wnioski

15: Przedstawienie finalnej wersji projektu i omówienie rezultatów. [2h] [30.01.23r.]

Literatura

[1] T. Windeatt, "Ensemble mlp classifier design," in *Computational Intelligence Paradigms*. Springer, 2008, pp. 133–147.

[2] A. Mechelli and S. Viera, *Machine learning: methods and applications to brain disorders*. Academic Press, 2019.

[3] D. T. Barus, R. Elfarizy, F. Masri, and P. Gunawan, "Parallel programming of churn prediction using gaussian naïve bayes," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2020, pp. 1–4.

[4] C. Kingsford and S. L. Salzberg, "What are decision trees?" *Nature biotechnology*, vol. 26, no. 9, pp. 1011–1013, 2008.

[5] T. Hothorn and B. Lausen, "Double-bagging: combining classifiers by bootstrap aggregation," *Pattern Recognition*, vol. 36, no. 6, pp. 1303–1309, 2003.

[6] M. Somvanshi, P. Chavan, S. Tambade, and S. Shinde, "A review of machine learning techniques using decision tree and support vector machine," in *2016 international conference on computing communication control and automation (ICCUBEA)*. IEEE, 2016, pp. 1–7.

[7] C. D. Sutton, "Classification and regression trees, bagging, and boosting," *Handbook of statistics*, vol. 24, pp. 303–329, 2005.

[8] M. P. Sesmero, J. A. Iglesias, E. Magán, A. Ledezma, and A. Sanchis, "Impact of the learners diversity and combination method on the generation of heterogeneous classifier ensembles," *Applied Soft Computing*, vol. 111, p. 107689, 2021.